COMPUTER IM SELBSTBAU

EINE PUBLIKATION VON CHIP

DAS MIKROCOMPUTER-FACHMAGAZIN

1. Auflage 1983
Alle Rechte, auch der Übersetzung , vorbehalten.
Kein Teil des Werkes darf in irgendeiner Form
(Druck, Fotokopie, Mikrofilm oder einem anderen
Verfahren) ohne schriftliche Genehmigung des
Verlages oder durch Verwendung elektronischer
Systeme vertrieben, vervielfältigt oder ver-
trieben werden.
Für Fehler im Text, in Schaltbildern, Aufbau-
skizzen, Stücklisten usw. die zum Nichtfunktionieren
oder evtl. zum Schadhaftwerden von Bauelementen
führen, kann keine Haftung übernommen werden.

Printed in Germany
Copyright 1983 by Vogel-Verlag Würzburg

25,-
BS
11b

INHALTSVERZEICHNIS

1.	Anmerkungen für den Zusammenbau	S 2
2.	Beschreibung der HARDWARE	S 3
2.1	Die CPU 8085-Grundplatine	S 4
2.1.1	Steuersignalerzeugung und Interrupt-verwendung	S 5
2.1.2	Lageplan der Bauteile	S 6
2.1.3	Bauteileübersicht	S 10
2.1.4	Steckerbelegung	S 12
2.2	Der Umgang mit dem Speicher	S 13
2.2.1	Aufbau und Arbeitsweise der 16K-Speicher-platine	S 13
2.2.2	Adressenaufteilung und Decodierung	S 15
2.2.3	Lageplan der Bauteile	S 19
2.2.4	Bauteileübersicht	S 20
2.3	Die RAM/VIDEO-Grundplatine	S 21
2.3.1	Adressierung der Schnittstellen-Bausteine 8255A	S 21
2.3.2	Anschluß einer ASCII-Tastatur	S 24
2.3.3	Die Drucker-Schnittstelle	S 28
2.3.4	Das Kasetten-Interface	S 29
2.3.4.1	Aufzeichnungsformat	S 31
2.3.5	Der VIDEO-Teil mit CRT 96364A	S 34
2.3.5.1	Der Zeichengenerator ROM 2513	S 35
2.3.5.2	Programmieren des Control-Decoders 74S287	S 36
2.3.6	Lageplan der Bauteile	S 37
2.3.7	Bauteileübersicht	S 38
2.3.8	Steckerbelegung	S 41
2.4	Ein Netzteil	S 43
2.4.1	Anschluß der Betriebsspannung	S 43
2.5	Inbetriebnahme des Systems	S 44
2.6	Erweiterung der HARDWARE	S 46

2.6.1	Anschluß einer LED-Anzeige	S 46
2.6.2	Anschluß einer Sieben-Segment-Anzeige	S 48
2.6.3	Eine EPROM-Programmiereinrichtung	S 49
2.6.4	Die serielle Schnittstelle 8251A	S 49
2.7	Speicheraufteilung	S 52
3.	Beschreibung einiger Bausteine	S 53
4.	Gesamtübersicht verwendeter Bausteine mit Anschlußbelegung	S 74
5.	Platinenvorlagen	S 93
6.	Beschreibung der SOFTWARE	S 101
6.1	Übersicht der Befehlseingabe	S 102
6.2	EDITIEREN eines Programms	S 104
6.2.1	Funktion einiger Steuertasten	S 106
6.3	EDITIEREN eines vorhandenen Programms	S 111
6.4	Aufruf des ASSEMBLERS	S 112
6.5	Speicher auslesen und beschreiben	S 117
6.6	Kommandos zum Softwaretest	S 117
6.7	Befehlseingabe für Recorder und Drucker	S 119
6.8	Steuerung einer EPROM-Programmiereinrichtung	S 120
6.9	Zusammenstellung systemeigener Unterprogramme	S 123
7.	Einige Programmierbeispiele	S 128
8.	Lösungen	S 129
9.	Bezugsquelle System-Software	S 131

©
Vogel-Verlag KG
Postfach 6740
8700 Würzburg 1

EINLEITUNG

Mit der vorliegenden Bauanleitung des CHIP-Computers soll der Zweck verfolgt werden, sich mit einem Microprozessor und seinen Peripheriebausteinen auf preiswerte Weise vertraut zu machen. Dieses Computersystem selbst zu bauen ist nicht nur kostengünstig, sondern wohl auch der beste Weg sich HARDWARE - u. SOFTWARE-Kenntnisse eines Microprozessors anzueignen.
Der CHIP-Computer bietet aufgrund seiner Konfiguration recht große Experimentiermöglichkeiten, die wohl wichtigste Eigenschaft dieses Systems ist der modulare Aufbau.
Zur raschen Einarbeitung in die spezifischen Eigenschaften des Microprozessors 8085 steht dem Anwender ein relativ komfortables 16K-Byte ROM-residentes Betriebssystem zur Verfügung. Unterstützt durch die bestehende, umfangreiche deutschsprachige Dokumentation des Microprozessors 8085, können mit dem CHIP-Computer durch eine Vielzahl von praktischen Übungsbeispielen grundlegende Kenntnisse in der ASSEMBLER-Sprache (8085/8080) erworben werden.
Die nachfolgende Systembeschreibung möchte nicht nur als reine Bauanleitung verstanden werden, sie wurde auch zu dem Zweck erstellt, dem Leser anhand konkreter Schaltungsunterlagen das Zusammenspiel von HARD - u. SOFTWARE eines 8085-Microcomputersystems in verständlicher Form aufzuzeigen.

München 1983 H.J. Blank

ANMERKUNG FÜR DEN ZUSAMMENBAU

Wie schon erwähnt, ist das ganze System modular aufgebaut, d.h. auf zwei durch ein BUS-Kabel miteinander verbundenen Grundplatinen, befinden sich steckbar die EPROM-u. RAM-Karten, sowie das VIDEO-Interface.
Nur durch die einwandfreie Funktion aller vorhandenen Systemkomponenten, kann der CHIP-Computer störungsfrei arbeiten bzw. überhaupt erst in Betrieb genommen werden. Dies bedeutet, daß mit größter Sorgfalt die einzelnen Platinen hergestellt und bestückt werden müssen; bedenken Sie bitte, der Ausfall oder Kurzschluß einer einzigen Leitungsführung kann das gesamte System zum Erliegen bringen.
Microcomputer sind sehr komplexe Schaltungen und entsprechend aufwendig ist dann die Suche nach den Fehlern.
Die in dieser Bauanleitung veröffentliche Platinenvorlagen sind mehrfach einem praktischen Test unterstellt worden, einige dieser Microcomputersysteme werden bereits erfolgreich seit längerer Zeit für Aus- u. Fortbildungszwecke eingesetzt.

MERKMALE DES CHIP-COMPUTERS

- Modulares Konzept
- ASCII-Tastatur zur Programmeingabe
- ANZEIGE auf Bildschirm 16 Zeilen/64 Zeichen
- Schrittweise Ablaufverfolgung
- Absolut oder symbolisch gesetzte Unterbrechungspunkte
- Leistungsfähiger EDITOR u. ASSEMBLER
- Verwendung symbolischer Adressen
- Eingebauter DISASSEMBLER zur Fehlersuche
- 16K-Byte ROM-Speicher mit Betriebssystem
- 32K-Byte RAM-Speicher als Arbeitsspeicher
- Kassetten-Interface für die Ein-u. Ausgabe von Daten
- Paralleles Interface für einen Drucker
- Erweiterbar mit serieller Schnittstelle
- Anschlußmöglichkeit einer EPROM-Programmiereinrichtung

BESCHREIBUNG DER HARDWARE

In den folgenden Kapiteln sollen dem Leser über den Zusammenbau hinaus einige Grundkenntnisse vermittelt werden, die erforderlich sind, um später selbst Microcomputer entwerfen und anwenden zu können.
Außerdem soll an bestimmten Stellen eine Begriffserklärung im Vordergrund stehen, um auch denjenigen Interessenten den gewünschten Erfolg zu geben, die vielleicht wirklich am bewußten Punkt Null anfangen müssen.
Wer einen Microprozessor erklären und beschreiben will, muß zunächst einmal sagen was ein Microcomputer ist, denn die Wirkungsweise eines Microprozessors kann eigentlich nur verstanden werden, wenn man ihn im Zusammenhang mit seinen ergänzenden Baugruppen sieht.
BILD 1 zeigt nun das Blockschaltbild des CHIP-Computers.
Wie man dort erkennt, bildet der Microprozessor 8085 die Zentraleinheit des Microcomputers. Die Signalübertragung innerhalb und außerhalb der CPU 8085 erfolgt über ein Leitungssystem, das allgemein BUS-System genannt wird. Über dieses BUS-System erfährt der Microprozessor WAS er tun soll, mit WEM er es tun soll und WOHIN er das Ergebnis schicken soll.

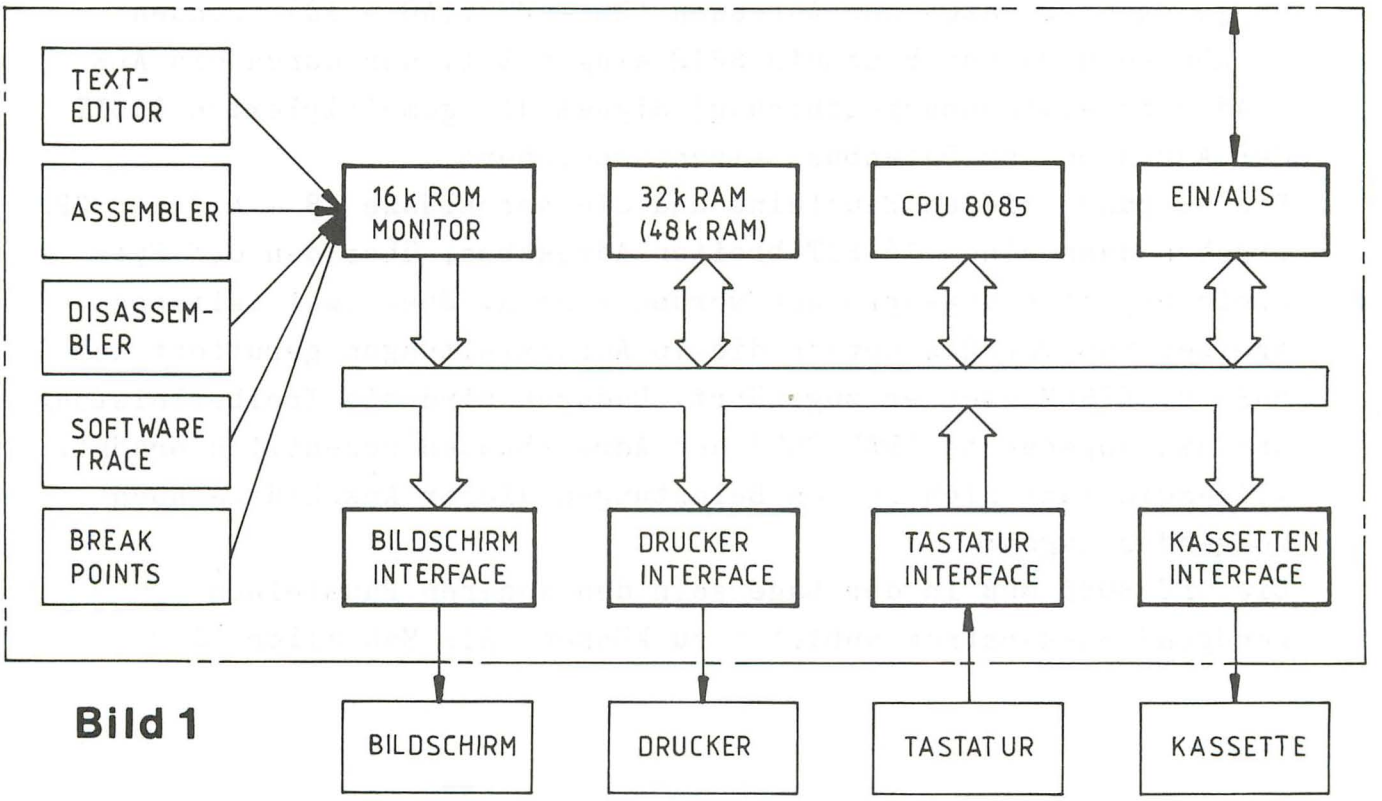


Bild 1

Der Microprozessor 8085 hat nun die Aufgabe das Zusammenwirken der einzelnen Systemkomponenten richtig zu steuern. Dafür benötigt der CHIP-Computer einen Festwertspeicher von 16K-Byte als Programmspeicher, über den der 8085 die Befehle und Daten bekommt, mit denen er die einwandfreie Funktion aller angeschlossenen Bauteile sicherstellt.
Dieses Betriebssystem (MONITOR) ist also nichts weiter als ein Verwaltungsprogramm, daß bestimmte Grundfunktionen,wie z.B. Eingabe Tastatur, Ausgabe Bildschirm, also die Behandlung der vorhandenen Schnittstellen sicherstellt.

2.1 DIE CPU 8085-GRUNDPLATINE

BILD 2 zeigt die Schaltung, die auf der CPU 8085-Grundplatine vorhanden ist. Hier sitzt das Herz des CHIP-Computers, der 8085-Microprozessor, der die Koordination aller Funktionsgruppen übernimmt und sämtliche interne Abläufe steuert (siehe auch 3.1).
Typisch für diese CPU, im Gegensatz zu seinem Vorgänger 8080, ist sein sogenannter "gemultiplexter" Datenbus AD0 - AD7.
Diese Busstruktur wurde deshalb gewählt, weil durch sie Anschlüsse des Bausteins frei wurden, was die Integration zusätzlicher Funktionen ermöglichte.
Um jetzt aber Daten und Adressen (D0 - D7),(A0 - A8) trennen zu können wird der Baustein 8212 eingesetzt, der durch ein ALE (Adressen-Zwischenspeicherung) Signal die gemultiplexten 8 BIT der Adressen vom Datenbus zwischenspeichert.
Die Ausgänge dieses Bausteins und die Anschlüsse A8 - A15 der CPU ergeben dann einen 16 BIT breiter Adressbus, über den 64K-Byte Speicherplätze angesprochen werden können. Über zwei Leitungstreiber Typ 74LS244 werden die 16 Adressleitungen gepuffert dem BUS- u. EPROM-Stecker zugeführt. Dadurch wird die Treiberleistung und das sogenannte "FAN OUT" des Adressbusses wesentlich erhöht.
Allgemein läßt sich zu den Belastungen dieser Anschlüsse noch folgendes sagen:
Die CPU 8085 muß in der Lage sein den anderen Bausteinen genügend Steuerstrom anbieten zu können. Als Maß aller Dinge

nimmt man eine TTL-Last, in der Regel gilt, daß der Ausgang des Microprozessors 8085 mindestens eine TTL-Last anzutreiben vermag. Im Gegensatz dazu benötigen LOW POWER SCHOTTKY-Schaltkreise einen viermal kleineren maximalen Eingangsstrom.
Es sei aber ergänzend darauf hingewiesen, daß neben den jeweiligen Steuerströmen, auch die Eingangskapazitäten umliegender Bausteine eine wesentliche Rolle spielen.
Bei der Größe des CHIP-Computers ist man deshalb gezwungen, die o.a. PUFFER-Verstärker in die Adressleitungen zu integrieren, was natürlich auch die Störsicherheit beträchtlich erhöht.

2.1.1 STEUERSIGNALERZEUGUNG U. INTERRUPTVERWENDUNG

Über DATEN, BEFEHLE u. ADRESSEN wird in den folgenden Kapiteln zu lesen sein, sie bilden sinnvoll aneinandergereiht die sogenannte SOFTWARE.
Für den Konstrukteur des CHIP-Computers dagegen, sind vor allem die Steuersignale sehr wichtig. Will die CPU 8085 z.B. DATEN an den Speicher abgeben, so signalisiert sie das an bestimmten Ausgängen den peripheren Bausteinen. Zu beachten ist, daß bei der Darstellung der STEUERSIGNALE die NEGATIV-LOGIK eine bedeutende Rolle spielt. Folgende TABELLE 1 soll die Aufbereitung der STEUERSIGNALE für den Speicher und den Ein/Ausgabe-Bausteinen des Systems näher verdeutlichen:

	$IO/\overline{M}$	$\overline{RD}$	$\overline{WR}$
SPEICHER LESEN	0	0	1
SPEICHER SCHREIBEN	0	1	0
EIN/AUSGABE LESEN	1	0	1
EIN/AUSGABE SCHREIBEN	1	1	0

IO=INPUT/OUTPUT, M=MEMORY, RD=READ, WR=WRITE

TABELLE 1 STEUERSIGNALE der CPU 8085

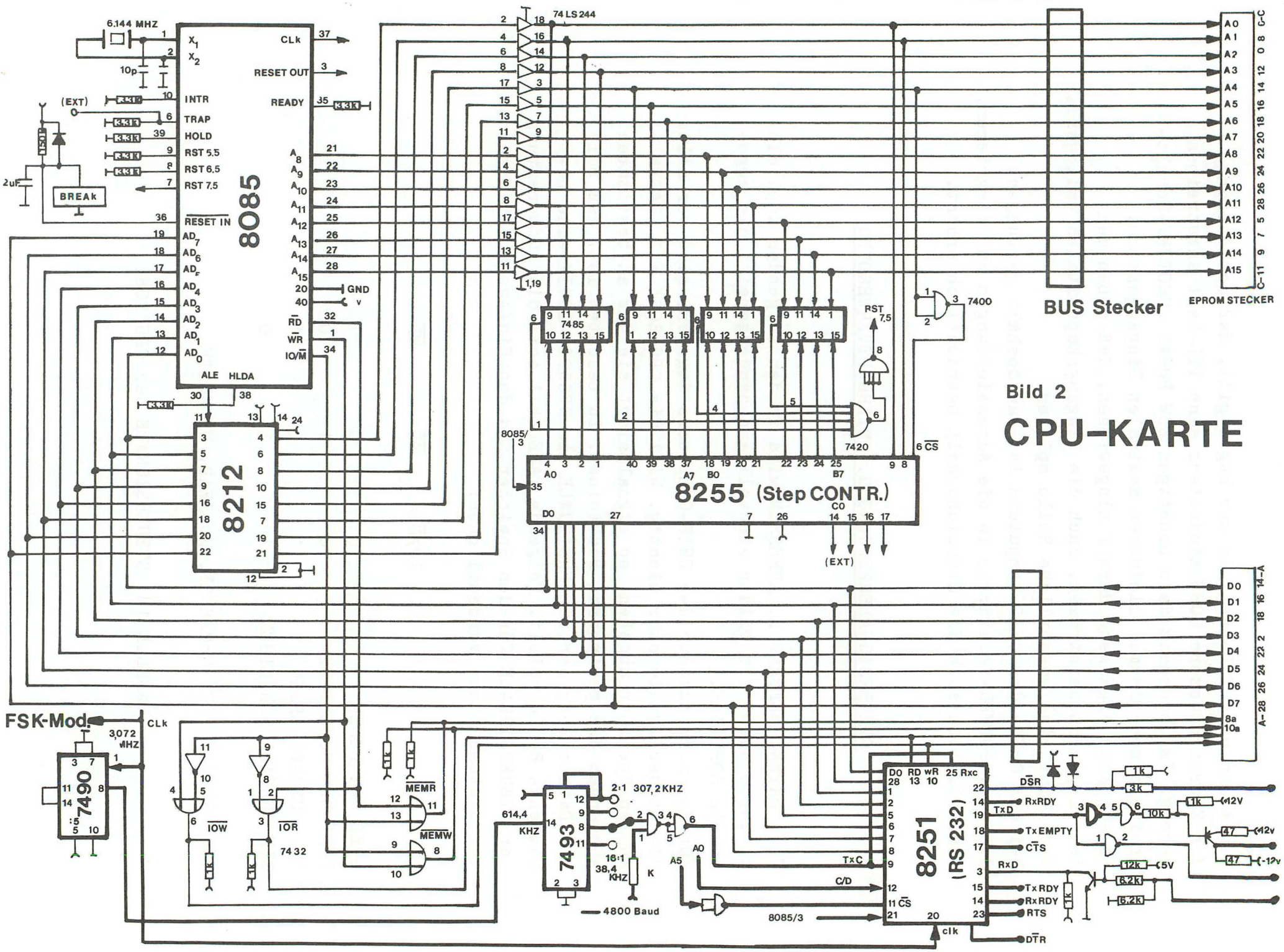


Bild 2

CPU-KARTE

Die auf dieser CPU-Grundplatine erzeugten Kontrollsignale sind für alle noch anzuschließenden Systemkomponenten von größter Bedeutung, deshalb werden sie ebenfalls zu den Steckeranschlüssen hingeführt.
Man vergegenwärtige sich noch einmal, daß der Datenverkehr für alle um den ADRESS- u. DATEN-BUS angesiedelten Bausteine über das gleiche Leitungsbündel, jedoch zu verschiedenen Zeiten abläuft. Die CPU 8085 hat also die Aufgabe diese STEUERSIGNALE so zu erzeugen, daß sie genau zum richtigen Zeitpunkt an dem angesprochenen Baustein anstehen.
Weiterhin ist in BILD 2 eine Schaltung mit dem Ein/Ausgabe-Baustein 8255A dargestellt, die bei einer späteren SOFTWARE-Entwicklung von größter Bedeutung ist. Sie ist die hardwaremäßige Realisierung einer umfangreichen PROGRAMM-Testhilfe.
Nach der Übersetzung eines Programms durch den ASSEMBLER, erfolgt in der Regel der Systemtest, d.h. die Funktion der vom Anwender erstellten SOFTWARE wird daraufhin getestet, ob das gewünschte Ziel erreicht worden ist oder nicht.
Mit dieser Schaltung gibt es nun die Möglichkeit, den Programmablauf nur schrittweise oder bis zu einer bestimmten Stelle durchführen zu lassen und das Ergebnis mit Hilfe des Microprozessors zu analysieren.
Durch den Baustein 8255A ergeben sich 24 programmierbare Anschlüsse, die in 3 Kanälen A,B u. C untergebracht sind (siehe 3.2). Die beiden Kanäle A und B können nun innerhalb des Betriebssystems so gesetzt werden, daß ein BIT-Vergleich über die Bausteine 74LS85 mit dem ADRESS-BUS hergestellt werden kann. Bei Gleichheit erzeugen die Ausgänge der 4-BIT-VERGLEICHER einen logischen "1" Impuls, der die UNTERBRECHUNG (RST 7.5) eines laufenden Maschinenprogramms ermöglicht (siehe auch 3.1).
In den Programmablauf des Microprozessors 8085 wird also von außen eingegriffen. Die CPU wird nun gezwungen in ein UNTERPROGRAMM zu springen, um eine bestimmte Funktion zu erfüllen, in unserem Fall die Darstellung sämtlicher REGISTER-Inhalte auf dem Bildschirm und Rückübersetzung des MASCHINENCODES (DISASSEMBLIERUNG).

Zu der Unterbrechungsverarbeitung läßt sich grundsätzlich noch folgendes sagen:
Normalerweise arbeitet der Microprozessor 8085 sein Programm in der Reihenfolge ab, wie es ihm bei der PROGRAMMIERUNG vorgegeben wurde. Bei einem so organisierten Programm müssen aber die angeschlossenen Geräte ihre Daten so lange bereit halten, bis die Abfrage ihrer Einheit an der Reihe ist. Beim 8085-Microcomputersystem gibt es nun die Möglichkeit, über eine relativ komfortable Logik, in den Programmablauf von außen einzugreifen und dadurch die CPU zu einem Sprung in ein ganz bestimmtes UNTERPROGRAMM zu veranlassen. Dieser Vorgang wird mit dem englischen Wort INTERRUPT bezeichnet, nähere Einzelheiten unter 3.1 .
Der untere Teil der in BILD 2 dargestellten Schaltung ist nur für eine Systemerweiterung SERIELLE SCHNITTSTELLE von Bedeutung, auf die in einem der folgenden Abschnitte eingegangen wird. Eine Bestückung ist deshalb nur bei Gebrauch notwendig.
Zu den verbleibenden Anschlüssen des 8085 läßt sich noch erwähnen, daß der TAKTGEBER im Baustein integriert ist und somit der 6.144 MHz Quarz direkt angeschlossen werden kann. Wobei zum einwandfreien Schwingen je ein Kondensator von 10pF gegen Masse geschaltet werden muß.
Über ein RC-Glied 150K/2.2uF wird sichergestellt, daß beim Einschalten der Spannung ein Rücksetz-Impuls an die CPU erzeugt wird, der PROGRAMM-ZÄHLER des Microprozessors geht auf die ADRESSE 0000H. Um jederzeit einen solchen Impuls herstellen zu können, wird eine BREAK-Taste oder ein Schalter ebenfalls benötigt. Dadurch wird gewährleistet, daß zu jeder Zeit ein Rücksprung in das BETRIEBSSYSTEM möglich ist. Nach einem Rücksetzen beginnt der 8085 mit der Ausführung der Befehle bei der Adresse 0000H und mit einem sogenannten gesperrtem Unterbrechungssystem. Diese INTERRUPT-Eingänge des 8085 dürfen bei Nicht-Verwendung keinesfalls freigelassen werden, sondern müssen mit PULL-DOWN-Widerständen (3,3K) auf Null-Potential gelegt werden.

Der Microprozessor 8085 unterscheidet also von sich aus zwischen SPEICHER und PERIPHERIE. Die in der TABELLE 1 von der CPU erzeugten STEUERSIGNALE benutzt eine Hilfsschaltung, um vier Kontrollsignale zu erzeugen, die mit dem STEUERBUS des Vorgängers 8080 übereinstimmen. BILD 3 zeigt diese Schaltung. Die Ausführung eines Programms besteht nun aus einer Folge von LESE- u. SCHREIB-Operationen, diese Operationen sind der einzige Datenaustausch zwischen dem Microprozessor 8085 und den übrigen Bausteinen des Systems. Sie reichen aus, um jeden BEFEHL oder jedes PROGRAMM auszuführen.

Anhand von TABELLE 1 erzeugt die in BILD 3 dargestellte Hilfsschaltung folgende Kontrollsignale:

	IO/$\overline{M}$	$\overline{RD}$	$\overline{WR}$ /	$\overline{MEMR}$	$\overline{MEMW}$	$\overline{IOR}$	$\overline{IOW}$
SPEICHER LESEN	0	0	1	0	1	1	1
SPEICHER SCHREIBEN	0	1	0	1	0	1	1
EIN/AUSGABE LESEN	1	0	1	1	1	0	1
EIN/AUSGABE SCHREIBEN	1	1	0	1	1	1	0

TABELLE 2 KONTROLLSIGNALE der Hilfsschaltung

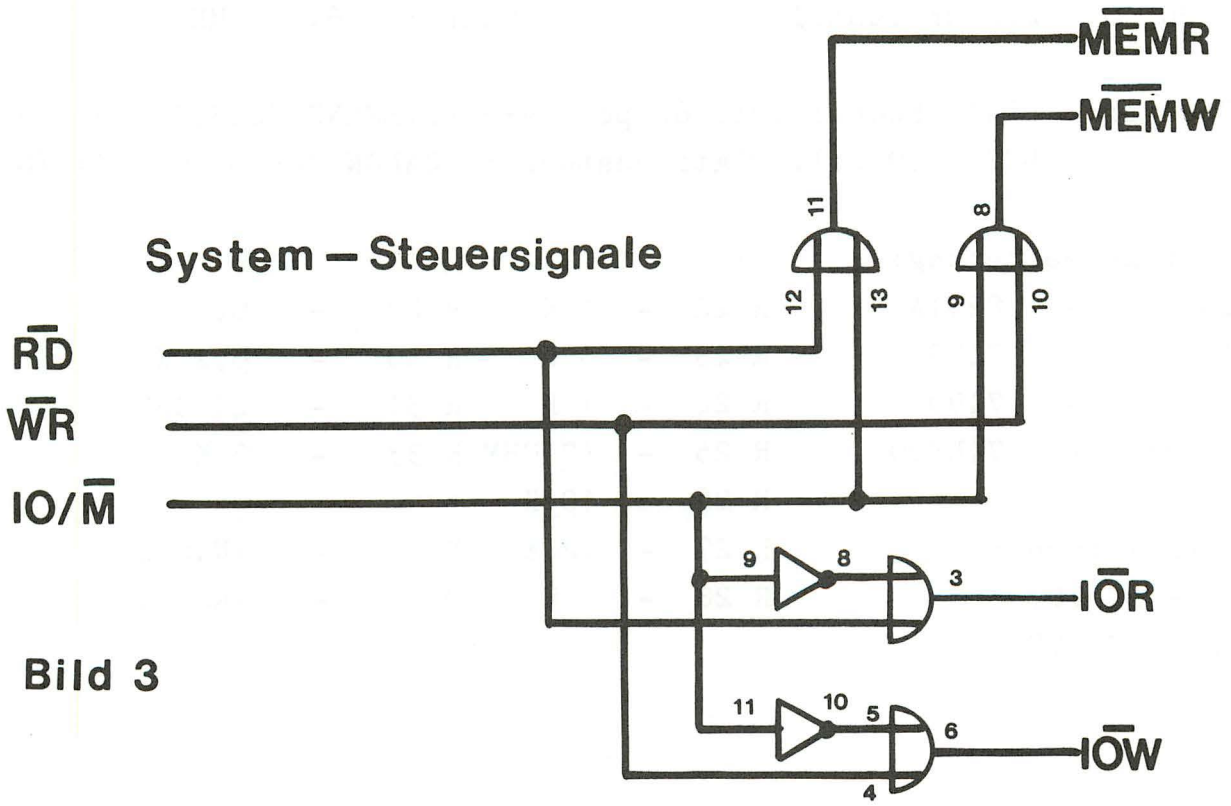


Bild 3

2.1.3 BAUTEILE-ÜBERSICHT CPU 8085-GRUNDPLATINE

Integrierte Schaltungen:

IC 20	-	8085A
IC 21	-	8212
IC 22	-	8255A
IC 26	-	74LS85
IC 27	-	74LS85
IC 28	-	74LS85
IC 29	-	74LS85
IC 30	-	74LS32
IC 31	-	74LS20
IC 32	-	74LS04
IC 35	-	74LS00
IC 40	-	74LS244
IC 41	-	74LS244

Widerstände: (Alle 1/4 W)

RR5.5	-	3,3 K
RTRA	-	3,3 K
RR6.5	-	3,3 K
RRST	-	3,3 K
RINT	-	3,3 K
RHOL	-	3,3 K
R 20	-	150 K
R 21	-	3,3 K
R 40	-	1 K
R 41	-	1 K
R 42	-	1 K
R 43	-	1 K

Kondensatoren:

C 11	-	10 pF
C 12	-	10 pF
C 20	-	2,2 uF Tantal

Dioden:

D 2= - 1N4004/oder ähnlich

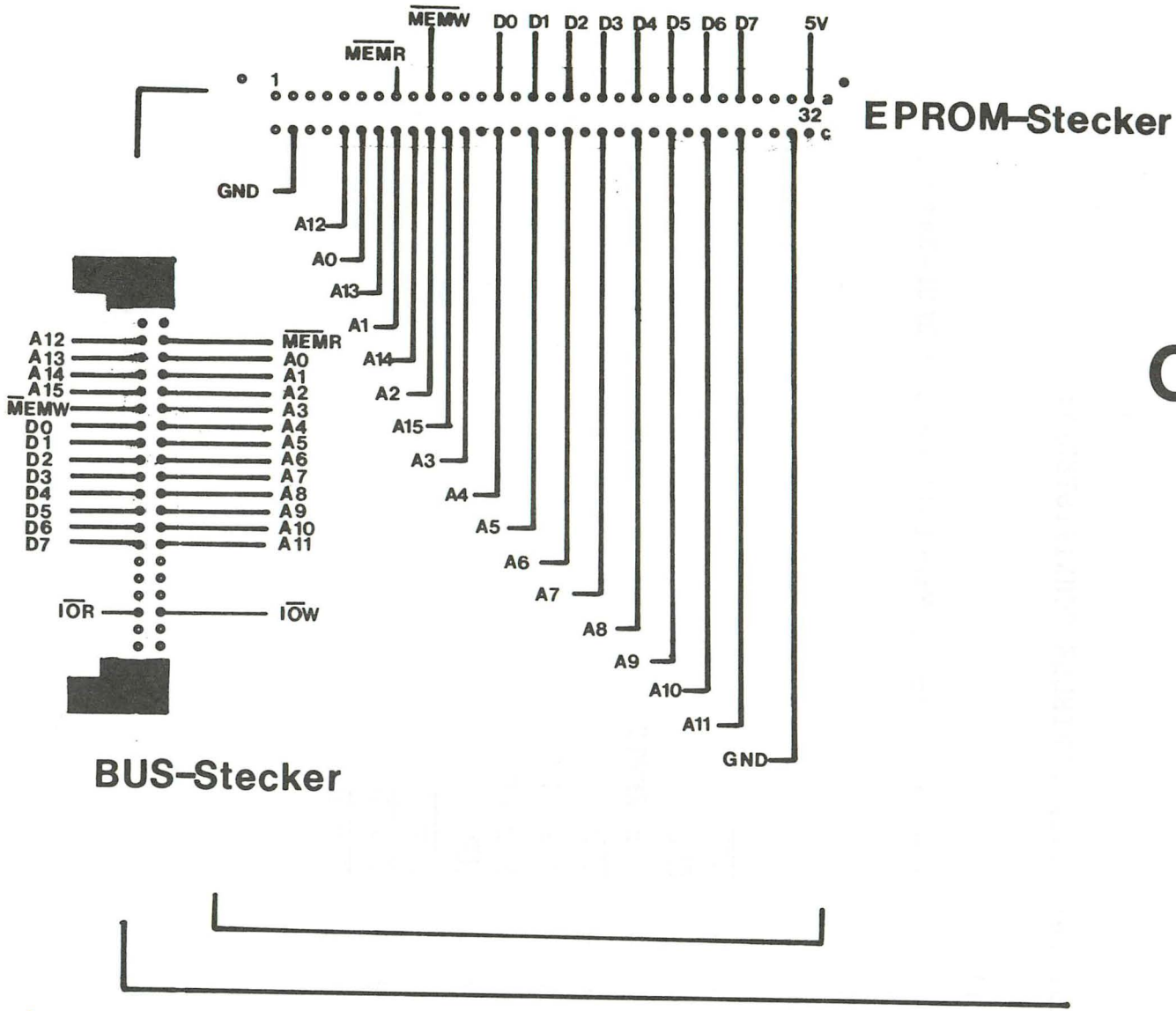
Quarz: 6.144 MHz

Stecker: STA1 Federleiste 64 pol. a+c (SIEMENS C42334 A192 A902)
BUS1 40 pol. Platinenstecker (CANON G08 D 40A3BKAA/075)

Systemerweiterung:

IC 23	-	8251A	R 22	-	1 K	R 29	-	6,2 K
IC 24	-	7493	R 23	-	3 K	R 30	-	6,2 K
IC 25	-	7490	R 24	-	1 K	R 31	-	47 OHM
IC 33	-	74LS00	R 25	-	47 OHM	R 33	-	3 K
			R 26	-	10 K			
			R 27	-	12 K	D 21	-	1N4004
			R 28	-	1 K	D 22	-	1N4004

Transistoren:
T1 - BC 107
T2 - BC 177



CPU PLATINE

Bild 4

2.1.4 STECKERBELEGUNG CPU 8085-GRUNDPLATINE

BILD 4 zeigt die Anschlußbelegung der EPROM 2716- Speicherkarte über den Stecker STA1, sowie die BUS-Verbindung über Stecker BUS1.
Nach dem LAGEPlAN der Bauteile (siehe 2.1.2) haben folgende Anschlüsse die Bedeutung:

+ RESET - Anschluss einer BREAK-Taste/Schalter
+ TRAP - INTERRUPT durch MONITOR auf ADRESSE 8300H gelegt.
+ RST6.5 - intern belegt.
+ S 15 - CLK-Signal der CPU für Kassetten-INTERFACE.

Systemerweiterung SERIELLE SCHNITTSTELLE:

+ S 11
+ S 12 ---- + S 16 (Einstellung der BAUD-Rate)
+ S 13
+ S 14

+ S 1 - $\overline{DTR}$
+ S 2 - $\overline{RTS}$
+ S 3 - T x EMPTY
+ S 4 - $\overline{CTS}$
+ S 5 - T x RDY
+ S 6 - R x RDY
+ S 7 - $\overline{DSR}$
+ S 8 - $\overline{R\ x\ D}$
+ S 9 - $\overline{T\ x\ D}$
+ S 10 - $\overline{T\ x\ D}$

2.2 DER UMGANG MIT DEM SPEICHER

Im vorangegangenen Abschnitt haben wir die Wirkungsweise des Microprozessors 8085 als Zentraleinheit (CPU) des CHIP-Computers etwas näher kennengelernt.
Er ist also in der Lage BEFEHLE und DATEN über sein BUS-System mit Hilfe von STEUERSIGNALEN einzuholen und zu verarbeiten.
Der 8085 gibt eine Adresse und ein Steuersignal aus und überträgt dann entweder Daten auf dem DATEN-BUS zu einem Speicher oder liest Daten von einem Speicher über den BUS ein.
Wie man sieht, werden im Speicher Befehle und Daten aufbewahrt.
BEFEHLE sind nun codierte Informationen, die das Verhalten des CHIP-Computers steuern, während DATEN codierte Informationen sind, die von der CPU 8085 verarbeitet werden.
Der Speicherbereich des CHIP-Computers läßt sich nun in zwei Hauptgruppen aufteilen:

1. Nur-Lese-Speicher EPROM, die im Betrieb nur gelesen werden können.
2. Schreib-Lese-Speicher RAM, in die auch geschrieben werden kann.

Dynamische RAM-Bausteine sind zwar kostenmäßig vorteilhafter, es taucht jedoch das Problem des Wiederauffrischens (REFRESH) auf, der nicht unabhängig von der CPU durchgeführt werden kann. Dieses Verfahren erfordert in der Regel den Einsatz von zusätzlichen Bausteinen und ist für den Selbstbau wesentlich komplizierter in der Handhabung und bei ev. Fehlersuche. Aus diesem Grund wurde im CHIP-Computer einem einfacheren und verständlichem Speicheraufbau der Vorzug gegeben.

2.2.1 AUFBAU U. ARBEITSWEISE DER 16K-BYTE-SPEICHERPLATINE

BILD 5 zeigt den Aufbau der 16K-Byte-Speichereinheit.
Charakteristisch für den Schaltungsaufbau ist die Tatsache, daß die Platine entweder mit EPROM- oder RAM-Speicherbausteinen bestückt werden kann.

Die Kompatibilität der verwendeten Bausteine 2716 und 6116 ermöglicht diesen relativ einfachen und übersichtlichen Speicheraufbau.
Der 2716 ist ein UV-löschbarer u. elektrisch programmierbarer FESTWERTSPEICHER für 16.384 BIT, organisiert mit 2048 ADRESSEN zu je 8 BIT, diese Organisation gilt ebenfalls für den verwendeten 6116 RAM-Speicherbaustein, wobei 8 Bausteine gleichen Typs eine 16K-Byte Speichereinheit bilden.
Jeder Baustein wird mit seinen acht Datenleitungen unmittelbar an den DATEN-BUS des CHIP-Computers angeschlossen. Die Ausgangstreiber dieser Speicher besitzen eine sogenannte TRI-STATE-Logik. Dieser dritte Zustand ist weder log."0" noch log."1", sondern dadurch gekennzeichnet, daß die Ausgänge sehr hochohmig sind und somit von außen jeden eingeprägten logischen Zustand annehmen können. Von diesem Verfahren wird auch bei später beschriebenen Bausteinen Gebrauch gemacht.
Zur Adressierung der 2048 BYTES benötigt jeder Baustein 11 ADRESS-Leitungen (2^{11}= 2048), die mit dem ADRESS-BUS A0-A10 des Microprozessors 8085 über Pufferverstärker 74LS244 verbunden sind. Wie man erkennt,ist die Größe des 16K-Byte Speichers aus einer Anzahl von acht einzelnen Speicherbausteinen (RAM oder ROM) aufgebaut.
Die Adressierung einer einzelnen Speicherzelle erfolgt dann in zwei Stufen. Der Microprozessor gibt auf dem ADRESS-BUS die ADRESSE aus. Aus dieser ADRESSE muß zunächst abgeleitet werden, welcher Speicherblock, also welcher Baustein (1.RAM,2.RAM...) adressiert wird. Es muß unbedingt darauf geachtet werden, daß niemals verschiedene Bausteine gleichzeitig ausgewählt werden können. Hierzu wird aus der ADRESSE ein Baustein-Auswahl-Signal ($\overline{CS}$) abgeleitet, das dem entsprechenden Steuereingang des Speicherbausteins zugeführt wird. Nur wenn an diesem Eingang der richtige Pegel (log."0") anliegt, wird die Adresse decodiert, und es können DATEN gelesen oder geschrieben werden.
Die Erzeugung der richtigen Bausteine-Auswahl-Signale ist also innerhalb des Speichersystems von entscheidener Bedeutung. Diese Zusammenhänge sollen zum besseren Verständnis und im Hinblick auf spätere Eigenentwicklungen noch näher erläutert werden.

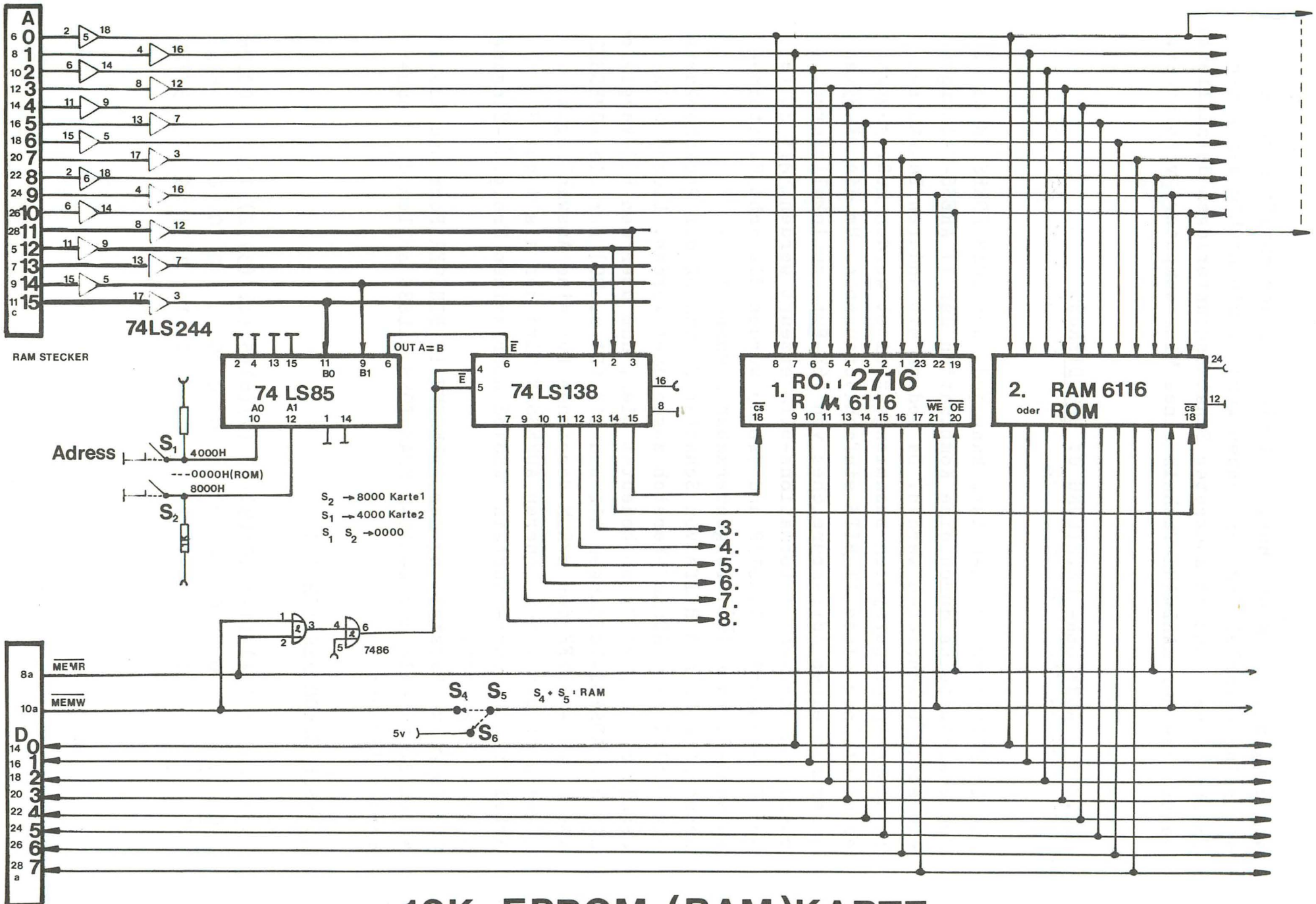


Bild 5 **16K EPROM (RAM)KARTE**

Wie eingangs schon erwähnt, soll diese Niederschrift nicht nur als reine Bauanleitung verstanden werden, sie möchte dem Leser darüber hinaus anhand konkreter Schaltungsunterlagen detaillierte Kenntnisse der HARDWARE-Technik eines Microcomputers vermitteln.

2.2.2 ADRESSEN-AUFTEILUNG U. DECODIERUNG

Wie in BILD 5 zu ersehen ist, kann jede der 2048 ADRESSEN eines Speicherbausteins durch eine Kombination der 11 ARESS-Leitungen eindeutig beschrieben werden. Da jedoch mit anderen Bausteinen in diesem Speichersystem zusammengearbeitet werden muß, genügt es nicht, nur diese 11 BITS einfach vorzugeben. Um die 2K-Bytes innerhalb eines Adressbereiches von 16K-Byte zu lokalisieren, müssen zusätzliche Informations-BITS erzeugt werden. Mit Hilfe der TTL-Bausteine 74LS138 und 74LS85 werden die acht Bausteine-Auswahl-Signale ($\overline{CS}$) folgendermaßen gewonnen:
Zunächst kann mit Hilfe von Schalter S1 und S2 der Speicherkarte ein definierter ADRESS-Bereich zugeordnet werden. Liegen beide Schalter auf log"0", so bedeutet dies, daß über den BIT-Vergleich mit den ADRESS-Leitungen A14 und A15 (B1 u. B0) nur die ADRESSEN von 0000H - 3FFFH des EPROM-Bereiches für eine Auswahllogik zugelassen sind. Bei höheren ADRESSEN wird der als 1 aus 8 arbeitende Decoder-Baustein 74LS138 über seinen Freigabe-Eingang (PIN 6) gesperrt.
Die folgende Darstellung der ADRESSEN in BINÄRER Form und die Wahrheitstabelle des 4-BIT-KOMPARATORS 74LS85 sollen dies näher verdeutlichen:

ADRESSE EPROM BEREICH 16K

(15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	0000H
$\underline{0}$	$\underline{0}$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	bis
$\underline{0}$	$\underline{0}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFFH

ADRESSE RAM BEREICH 16K (I) S2 = log."1" , S1 = log."0"

(15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000H bis
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	7FFFH

ADRESSE RAM BEREICH 16K (II) S2 = log"0" , S1 = log."1"

(15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000H bis
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	BFFFH

WAHRHEITSTABELLE 74LS85

A0 = B0 u. A1 = B1 - 0 - bedeutet Freigabe von 74LS138
A0 = B0 u. A1 ≠ B1 - 1
A0 ≠ B0 u. A1 = B1 - 1
A0 ≠ B0 u. A1 ≠ B1 - 1

BILD 5 DARSTELLUNG DER SPEICHERADRESSEN

Mit der in BILD 5 dargestellten DECODIER-Logik der ADRESSEN wird nun sichergestellt, daß eine 16K-blockweise Auswahl der drei Speicherkarten mit Hilfe eines Schalters oder von Lötbrücken möglich ist.
Nun braucht man nur noch dafür zu sorgen,daß immer nur einer der acht Bausteine mit Hilfe von $\overline{CS}$-Signalen (CHIP-SELECT) ausgewählt wird. Diese Aufgabe wird von dem schon erwähnten Baustein 74LS138 folgendermaßen durchgeführt:
Als Beispiel soll hier die BINÄRE Darstellung der Speicheradressen des EPROM-Bereiches herangezogen werden, wobei für den RAM-Adressbereich Analoges gilt.

1. EPROM 2716 (2K x 8)

(15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000H bis
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	07FFH

2. EPROM 2716 (2K x 8)

(15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0800H
																bis
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0FFFH

3. EPROM 2716 (2K x 8)

(15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000H
																bis
0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	17FFH

4. EPROM 2716 (2K x 8)

(15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1800H
																bis
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFFH

-

-

-

8. EPROM 2716 (2k x 8)

(15)	(14)	(13)	(12)	(11)	(10)	(9)	(8)	(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	
0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	3800H
																bis
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFFH

BILD 6 DARSTELLUNG DER EPROM-SPEICHERADRESSEN

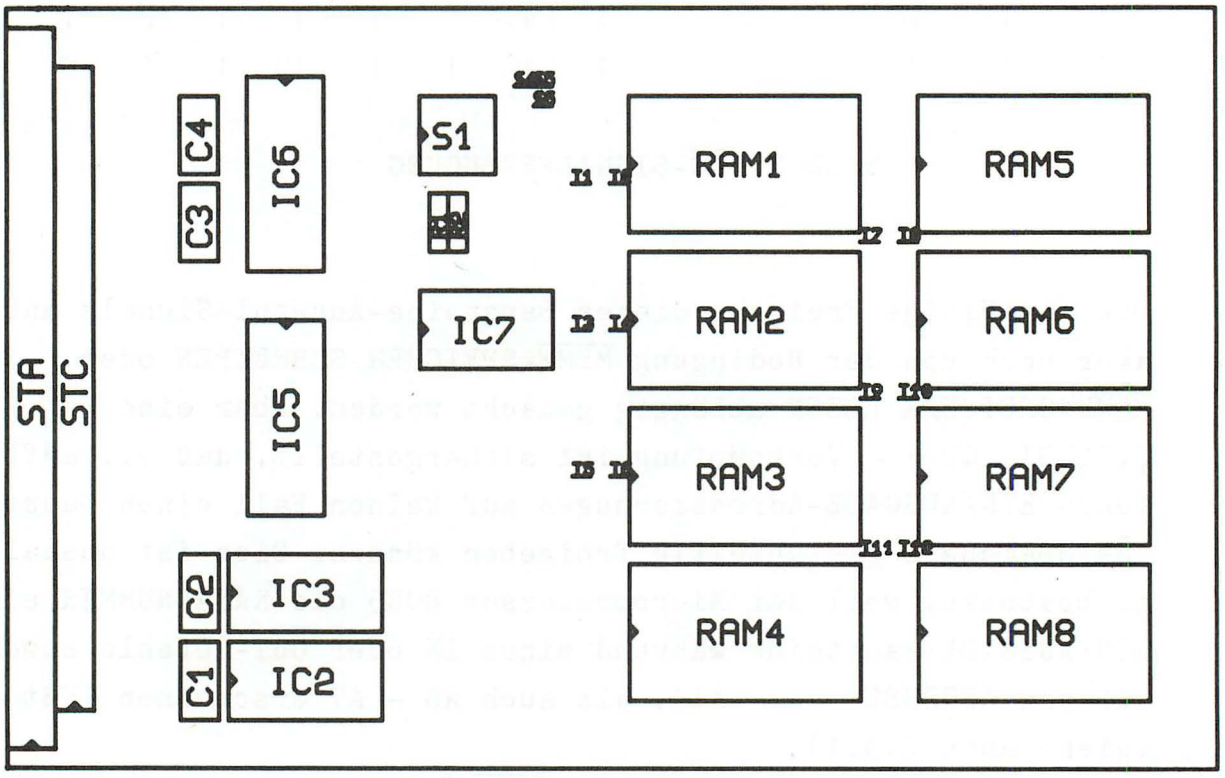
Wie man aus BILD 6 und BILD 7 erkennen kann, wird mit Hilfe der drei Adressleitungen A11 bis A13 sichergestellt, daß immer nur ein Baustein des Speichersystems über das $\overline{CS}$-Signal ausgewählt werden kann.

EINGÄNGE 74LS138:			AUSGÄNGE 74LS138:							
A13	A12	A11	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0

BILD 7 $\overline{CS}$-SIGNALERZEUGUNG

Die endgültige Freigabe dieses Bausteine-Auswahl-Signals muß aber noch von der Bedingung $\overline{MEMW}$=SPEICHER SCHREIBEN oder $\overline{MEMR}$=SPEICHER LESEN abhängig gemacht werden. Über eine EXCLUSIV ODER - Verknüpfung ist sichergestellt, daß ev. auftretende EIN/AUSGABE-Adressierungen auf keinen Fall einen Baustein des Speichers gleichzeitig freigeben können. Dies ist deshalb zu beachten, weil der Microprozessor 8085 die KANALNUMMER eines EIN/AUSGABE-Bausteins während eines IN oder OUT-Befehls sowohl auf den ADRESSEN A8 - A15, als auch A0 - A7 erscheinen läßt (siehe auch 2.3.1).
Der auffälligste Unterschied in der Organisation des RAM-Bausteins 6116 im Vergleich zum EPROM 2716 besteht darin, daß wegen der hinzu gekommenen Schreibfunktion der Datenweg nicht mehr den Charakter eines Ausgangs hat, sondern wahlweise als Ein- oder Ausgang wirkt
Deshalb findet man außer den BIDIREKTIONALEN Datenleitungen noch einen $\overline{WE}$ (WRITE ENABLE)-Anschluß vor. Der logische Pegel auf dieser SCHREIB-FREIGABE-Leitung legt fest, ob der Speicher 6116 im LESE oder SCHREIB- Betrieb arbeiten soll.

2.2.3 LAGEPLAN DER BAUTEILE 16K-SPEICHER-PLATINE



2.2.4 BAUTEILE-ÜBERSICHT 16K SPEICHERPLATINE

Integrierte Schaltungen:

IC 2 - 74LS85
IC 3 - 74LS138
IC 5 - 74LS244
IC 6 - 74LS244
IC 7 - 74LS86

RAM1 - 6116/2716(1) arbeitet als EPROM-Karte: S5+S6 BRÜCKE
RAM2 - 6116/2716(2) arbeitet als RAM-Karte : S4+S5 BRÜCKE
RAM3 - 6116/2716(3)
RAM4 - 6116/2716(4)
RAM5 - 6116/2716(5)
RAM6 - 6116/2716(6)
RAM7 - 6116/2716(7)
RAM8 - 6116/2716(8)

Widerstände: (Alle 1/4 W)

R1 - 1K
R2 - 1K

Schalter:

S1 - 4 pol. DUAL-IN-LINE
- oder Lötbrücke

Kondensatoren:

C1 - 1nF
C2 - 6,8uF Tantal
C3 - 1nF
C4 - 6,8uF Tantal

T1/T2 - 2.2uF Tantal
T3/T4 - 3.3nF
T5/T6 - 2.2uF Tantal
T7/T8 - 2.2uF Tantal
T9/T10 - 3,3nF
T11/T12- 2,2uF Tantal

Stecker: Belegung siehe BILD 4

STA Messerleiste 64 pol. a+c (SIEMENS 42334 A191 A514)

2.3 DIE RAM/VIDEO-GRUNDPLATINE

Diese zweite Grundplatine ist nun so konzipiert worden, um einmal eine steckbare Aufnahme des 32K-Byte RAM-Speichers zu ermöglichen, zum anderen sind hier alle noch benötigten SCHNITTSTELLEN (INTERFACES) des CHIP-Computers angeschlossen. Über eine 40 pol. Flachbandkabel wird die Verbindung zur CPU 8085-GRUNDPLATINE hergestellt (BILD 8).
Der 8085 verarbeitet Daten in digitaler Form, d.h. er übernimmt Daten in Form von Rechtecksignalen und sendet Daten in Form von Rechtecksignalen. Da diese Rechtecksignale vom Anwender eines Microcomputers jedoch nicht ohne DECODIERUNG verstanden werden können, kommt den EIN/AUSGABE-Schnittstellen eine ganz besondere Bedeutung zu.
Mit ihnen ist es dem Anwender nun möglich, Geräte anzuschließen, über die Daten in einer für dem Menschen verständlichen Form als Ziffer, Zeichen oder Bilder in den Microcomputer eingegeben bzw. in eben derselben Form vom Microcomputer ausgegeben werden können. Als EINGABE- bzw. AUSGABE-Daten kommen nicht nur Zahlen, Wörter oder Meßwerte in Betracht, sondern auch die verschiedenen Zustandsmeldungen, Signale von Tastaturen in unterschiedlicher Codierung und viele andere.
Der hier eingesetzte SCHNITTSTELLEN-Baustein 8255A ist nun dazu geeignet, die PARALLELE Übertragung von Daten zwischen dem Microprozessor 8085 und den Periphergeräten innerhalb des CHIP-Computers zu ermöglichen.

2.3.1 ADRESSIERUNG DER SCHNITTSTELLEN-BAUSTEINE 8255A

Auf die gesamte Vielfältigkeit dieses Bausteins kann in dieser Schrift zwar nicht eingegangen werden, er wird in seiner endgültigen Funktion erst durch das PROGRAMM eingestellt, doch es soll anhand der vorliegenden Schaltungsunterlagen die

Bausteine-Auswahl noch näher erläutert werden. Innerhalb des Aufbaus eines 8085-Microcomputersystems spielt der Anschluß von Peripherie-Bausteinen eine bedeutende Rolle.
Wie in BILD 3 dargestellt, unterscheidet man im CHIP-Computer mit einer Hilfsschaltung zwischen Speicher und Ein/Ausgabe-Bausteinen. Wenn der Microprozessor 8085 einen BEFEHL erhalten hat, der einen Datenaustausch mit einem 8255A-Baustein anordnet, dann sendet diese Schaltung STEUERSIGNALE aus, die den Speicher inaktiv lassen, dagegen aber den 8255A aktivieren. Die Zahl, die der Microprozessor dann auf den ADRESS-BUS gibt, wird nicht als Speicheradresse aufgefaßt, sondern als eine sog. KANALNUMMER; diese Zahl ist im Gegensatz zu den Speicheradressen nur 1 BYTE lang. Dadurch ergibt sich die Möglichkeit neben einer 64k-Byte Speicheradressierung noch bis zu 2^8=256 EIN/AUSGABE-Kanäle ansprechen zu können. Diese KANALNUMMER erscheint während eines EINGABE- oder AUSGABE-BEFEHLS in gespiegelter Form auf den Adress-Bus, d.h. sowohl an den Anschlüssen A0-A7, als auch A8-A15.
Die beiden STEUERSIGNALE $\overline{IOR}$ und $\overline{IOW}$ entkoppeln also den 8255A-Datenverkehr von den Speicherzugriffen. Jetzt braucht man nur noch dafür zu sorgen, daß die Bausteine-Auswahl in der richtigen Weise vorgenommen wird. Auch hier gelten dieselben Überlegungen wie bei der schon beschriebenen Speicherauswahl-Logik, daß verschiedene Bausteine auf keinen Fall gleichzeitig ausgewählt werden dürfen.
Im BILD 8 erkennt man deshalb, daß die $\overline{CS}$-Eingänge beider Bausteine an unterschiedlichen ADRESS-Leitungen liegen,(A2 u. A3). Die Verwendung eines ADRESS-BITS wird auch als eine sogenannte LINEARE Auswahl bezeichnet, während über die beiden ADRESS-Leitungen A0 und A1 der betreffende KANAL A,B oder C des Bausteins 8255A ausgewählt wird, der mit dem DATEN-BUS zum LESEN oder SCHREIBEN von DATEN verbunden werden soll.

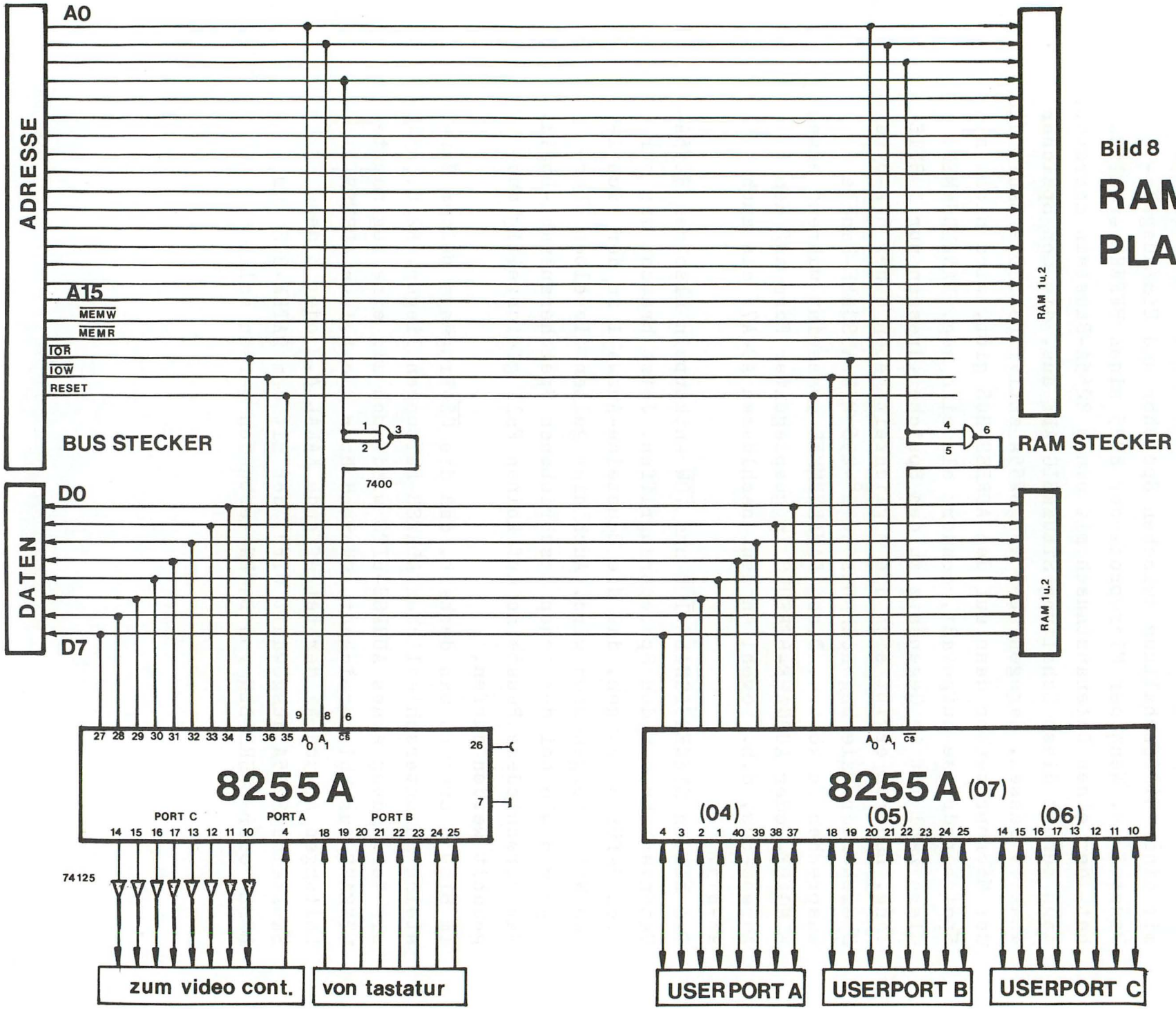


Bild 8

RAM/VIDEO PLATINE

Mit der folgenden Darstellung der prinzipiellen BETRIEBSARTEN des Bausteins 8255A wird die Festlegung der unterschiedlichen KANALNUMMERN deutlicher:

A1	A0	$\overline{IOR}$	$\overline{IOW}$	CS(A2) über 74LS00-$\overline{CS}$	CS(A3)	EINGABE/AUSGABE (LESEN) (SCHREIBEN)	8255(I)	8255(II)
0	0	0	1	1	0	KANAL A - DATENBUS	04H	--
0	0	0	1	0	1	KANAL A - DATENBUS	--	08H
0	1	0	1	1	0	KANAL B - DATENBUS	05H	--
0	1	0	1	0	1	KANAL B - DATENBUS	--	09H
1	0	0	1	1	0	KANAL C - DATENBUS	06H	--
1	0	0	1	0	1	KANAL C - DATENBUS	--	0AH
0	0	1	0	1	0	DATENBUS - KANAL A	04H	--
0	0	1	0	0	1	DATENBUS - KANAL A	--	08H
0	1	1	0	1	0	DATENBUS - KANAL B	05H	--
0	1	1	0	0	1	DATENBUS - KANAL B	--	09H
1	0	1	0	1	0	DATENBUS - KANAL C	06H	--
1	0	1	0	0	1	DATENBUS - KANAL C	--	0AH
1	1	1	0	1	0	STEUERWORT 8255(I)	07H	--
1	1	1	0	0	1	STEUERWORT 8255(II)	--	0BH

(A7)	(A6)	(A5)	(A4)	(A3)	(A2)	(A1)	(A0)	KANALNUMMER	
0	0	0	0	0	1	0	0	04H	
0	0	0	0	1	0	0	0	08H	
0	0	0	0	0	1	0	1	05H	
0	0	0	0	1	0	0	1	09H	
0	0	0	0	0	1	1	0	06H	
0	0	0	0	1	0	1	0	0AH	
0	0	0	0	0	1	1	1	07H	STEUERWORTAUSGABE
0	0	0	0	1	0	1	1	0BH	STEUERWORTAUSGABE

BILD 9 BETRIEBSARTEN-AUSWAHL 8255A

2.3.2 ANSCHLUSS EINER ASCII-TASTATUR

Da es für den Benutzer des CHIP-Computers extrem unbequem ist, Informationen in Form von BIT-Mustern bereitzustellen, sollen vom Sytem noch die Zeichen, der vom Anwender verwendeten Sprache gelesen und ausgegeben werden können. Dabei muß jedes Zeichen durch ein definiert zugeordnetes BIT-Muster bereitgestellt werden, im Falle des 8-BIT-Microprozessors 8085 ist es ja möglich in einem Speicher-Byte bis zu 256 verschiedene Zustände darzustellen.
Der hier verwendete 7-BIT-ASCII-Code überträgt nun die Information mit Hilfe digitaler Signale, wobei das höchstwertige BIT 2^7 nicht benutzt wird und die restlichen BIT gemäß folgender Tabelle den einzelnen ZEICHEN zugeordnet sind:

HEX	ASCII (STANDARD CODE FOR INFORMATION INTERCHANGE)		
41H	A	55H	U
42H	B	56H	V
43H	C	57H	W
44H	D	58H	X
45H	E	59H	Y
46H	F	5AH	Z
47H	G	2CH	,
48H	H	5BH	(
49H	I	5DH	)
4AH	J	u.weitere SONDER-Zeichen	
4BH	K	30H	0
4CH	L	31H	1
4DH	M	32H	2
4EH	N	33H	3
4FH	O	34H	4
50H	P	35H	5
51H	Q	36H	6
52H	R	37H	7
53H	S	38H	8
54H	T	39H	9

TABELLE 3 HEX-DARSTELLUNG EINIGER ASCII-ZEICHEN

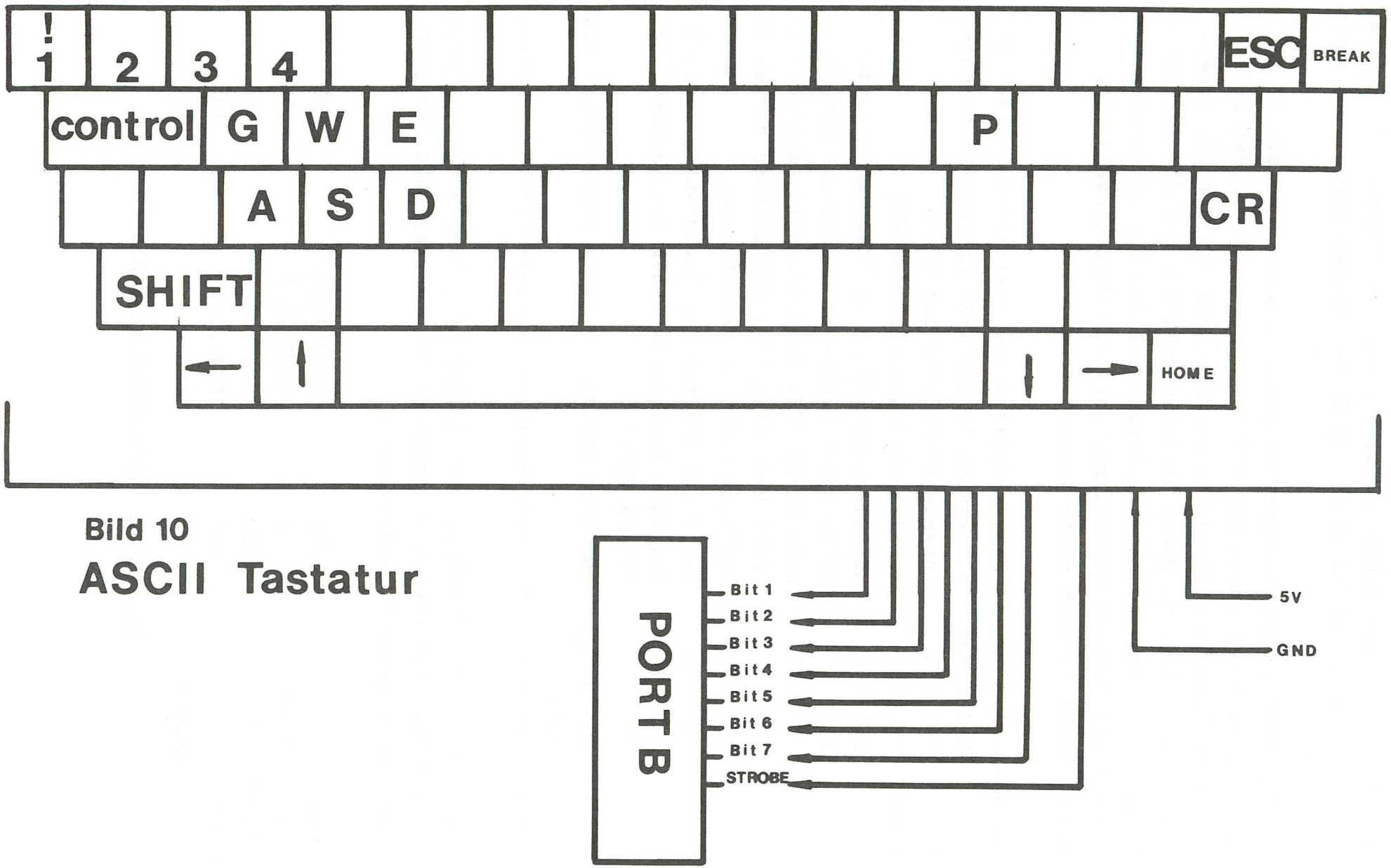


Bild 10

ASCII Tastatur

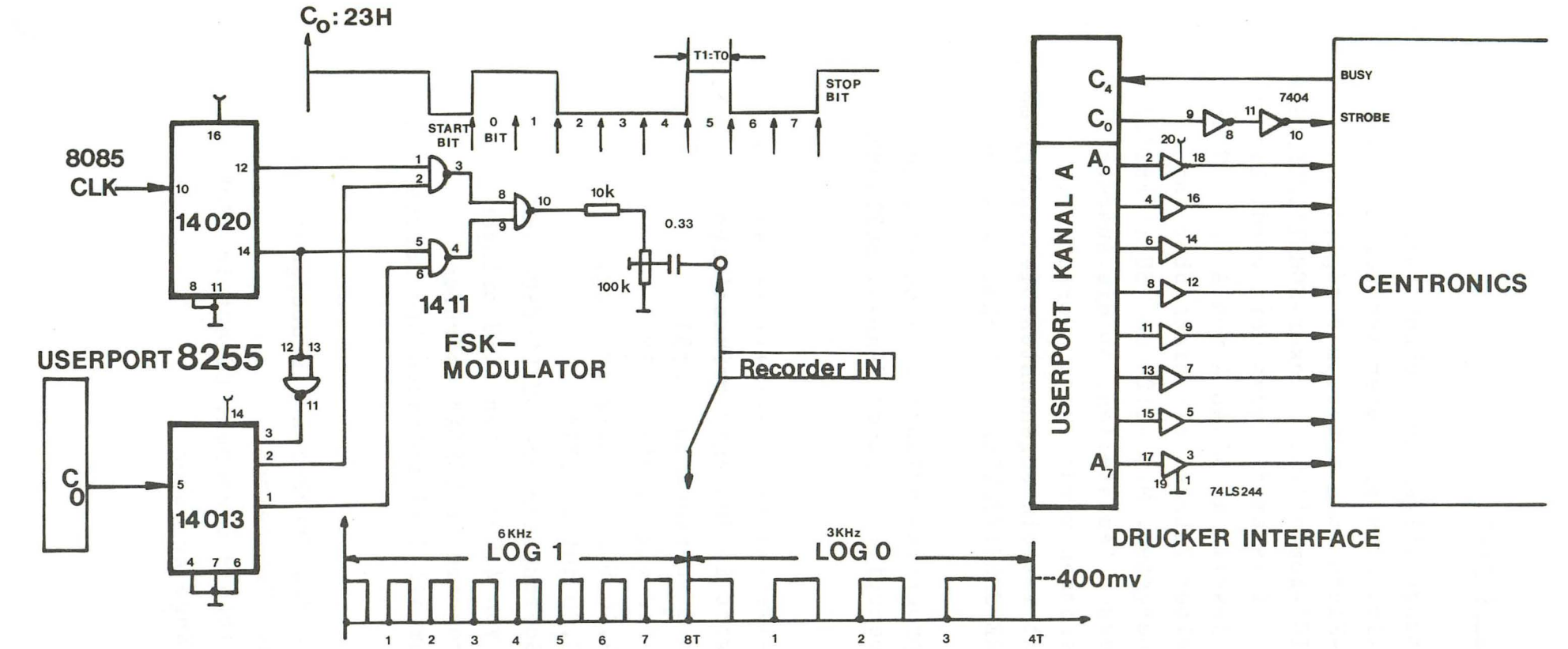
In BILD 10 ist nun dargestellt, wie eine beliebige ASCII-Tastatur über den PORT B des 8255A an den CHIP-Computer angebunden werden kann.
Der STROBE(Übernahme)-Impuls an dem höchstwertigen BIT B7 des Kanals B gibt mit einem log."1" Pegel die Meldung, daß alle sieben ASCII-BITS vollständig anstehen und von der CPU eingelesen werden können. Diese Steuerinformation muß die Tastaturlogik nach jedem Tastendruck bereitstellen. Die softwaremäßige Abfrage des Tastaturanschlusses soll die Wirkungsweise des STROBE-Impulses noch näher verdeutlichen:

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+        MVI A,92H   Steuerwort in den AKKU bringen(siehe auch 3.2)
+        OUT OBH     Ausgabe Steuerwort zum 8255A
+ TAST   IN 09H      Kanal B einlesen (ASCII-DATEN)
+        MOV B,A     Eingelesenes BYTE nach REGISTER B retten
+        RLC         Prüfe durch LINKS-Verschieben,ob STROBE="1"
+        JNC TAST    Wenn CARRY="0", dann nochmals einlesen
+        MOV A,B     CARRY ist jetzt "1", ASCII-Daten zurück nach A
+        ANI 7FH     MASKE,um BIT 7 zu selectieren
+        CPI ODH     Nun folgt Abfrage der Tastenwerte: RETURN ?
+        JZ RET      Ja, dann springe zu einem Programmteil RET
+        CPI OAH     CURSOR UNTEN ?
+        JZ CURS_D   JA, dann springe zu einem Programmteil CURS_D
+        usw.
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Außer Zahlen und Zeichen, wie in TABELLE 3 dargestellt, müssen von der Tastatur aber noch folgende STEUERFUNKTIONEN dem CHIP-Computer mitgeteilt werden können:

RETURN	ODH	TABULATOR	OFH
CURSOR UNTEN	OAH	CONTROL E	05H
CURSOR OBEN	OBH	CONTROL P	10H
CURSOR LINKS	08H	CONTROL S	13H
CURSOR RECHTS	09H	ESC	1BH

TABELLE 4 ASCII-STEUERZEICHEN


C0: 23H
T1:T0
STOP BIT
START BIT
0 BIT
1
2
3
4
5
6
7
8085 CLK
14 020
14 11
FSK-MODULATOR
10k
0.33
100k
Recorder IN
USERPORT 8255
C0
14 013
C4
C0
A0
A7
USERPORT KANAL A
BUSY
STROBE
7404
74LS244
CENTRONICS
DRUCKER INTERFACE
6KHz
LOG 1
3KHz
LOG 0
400mv
8T
4T

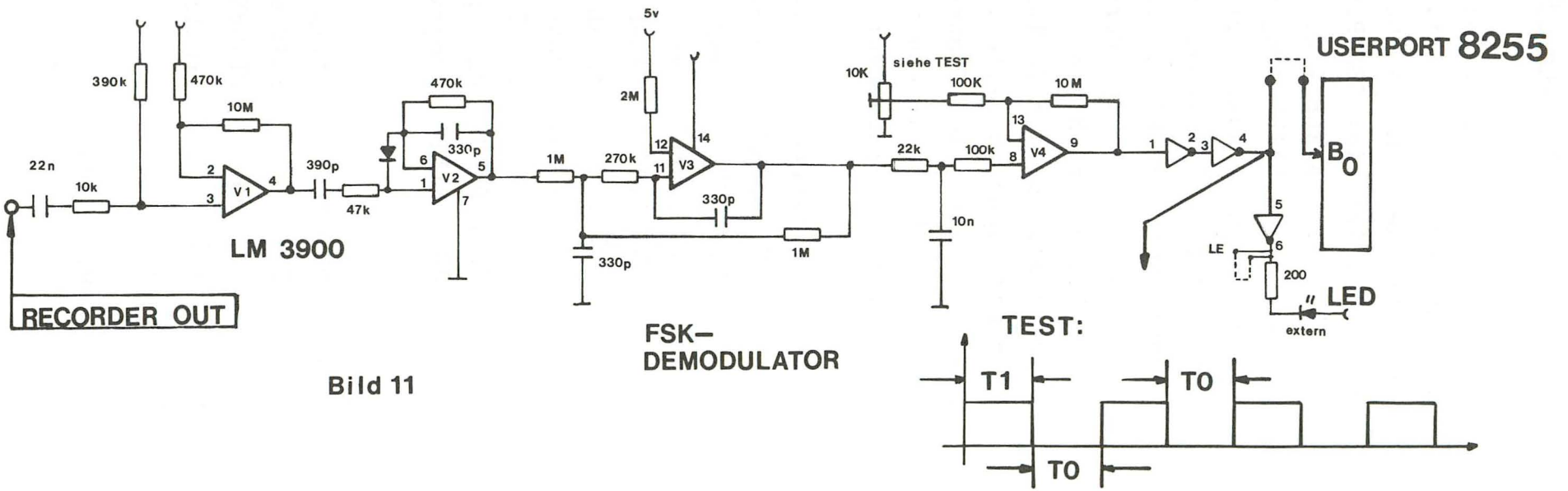

390k
470k
10M
22n
10k
V1
LM 3900
390p
47k
470k
330p
V2
1M
270k
5v
2M
V3
330p
1M
330p
10K
siehe TEST
100K
10 M
22k
100k
V4
10n
USERPORT 8255
B0
LE
200
LED
extern
RECORDER OUT
FSK-DEMODULATOR
TEST:
T1
T0
T0

Bild 11

2.3.3 DIE DRUCKER-SCHNITTSTELLE

Da der 8255A ein sehr leistungsfähiger und leicht verständlich aufgebauter Baustein ist, wird auch hier über ihn der Anschluß eines Druckers an das 8085-Microcomputersystem ermöglicht.
BILD 11 zeigt eine CENTRONICS-kompatible Drucker-SCHNITTSTELLE, die über die KANÄLE A u. C softwaremäßig gesteuert wird. Die auszugebenden DATEN liegen jeweils am KANAL A von A0 bis A7 an und werden über Ausgangstreiber 74LS244 mit einem Übernahme-Impuls der DRUCKER-LOGIK übergeben. Mit Hilfe des BUSY(FERTIG)-Signals kann der Microprozessor feststellen, ob die Ausgabe des nächsten Zeichens vorgenommen werden kann oder nicht.
Auch diesem Peripherie-Gerät ist ein sogenanntes Dienstprogramm zugeordnet, daß die SOFTWARE-SCHNITTSTELLE zwischen Gerät und Microprozessor verwaltet.
Die Einfachheit des Steuerprogramms erlaubt es auch an dieser Stelle dem Leser das Zusammenspiel zwischen HARD-u. SOFTWARE noch besser aufzuzeigen:

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+       MVI A,88H   Steuerwort in den AKKU bringen(siehe auch 3.2)
+       OUT 07H     Ausgabe Steuerwort zum 8255A
+       MVI A,FFH   Setze AKKU mit Konstante FF
+       OUT 04H     KANAL A,A0 bis A7 alles auf log."1"
+       OUT 06H     KANAL C,C0 auf log."1"
+       LXI H,4400H Erste Speicheradresse QUELL-PROGRAMM
+       MOV A,M     Inhalt von 4400H in den AKKU bringen
+       CALL DRUCK  Unterprogramm "DRUCKEN" aufrufen
+       JMP PRINTO  Programmteil TEXT abfragen und Source-Ende
+           -
+           -
+ DRUCK OUT 04H     DATEN an die Drucker-Logik übergeben
+       SUB A       AKKU auf NULL setzen
+       OUT 06H     STROBE Impuls erzeugen, C0 auf log."0"
+       NOP         Impulslänge erzeugen
+       NOP
+       NOP
+       MVI A,0FH
```

```
+        OUT 06H    STROBE Impuls wieder auf log."1" (CO)
+ WARTE IN 06H      BUSY Signal abfragen über C4
+        ANI 10H    BIT 2^4 selectieren, Rest interessiert nicht
+        CPI 10H    Ist BUSY-Signal auf log."1" ?
+        JZ WARTE   Ja, dann gehe in Warteschleife
+        RET        Nein, dann springe zum Hauptprogramm u. hole
+                                    neue DATEN
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

2.3.4 DAS KASSETTEN-INTERFACE

Da beim Abschalten des CHIP-Computers das erstellte Programm verloren geht, ist es sinnvoll einen externen DATEN-SPEICHER anschließen zu können. Hier bieten sich gerade Audio-Kassettengeräte wegen ihres niedrigen Preises an, es können damit PROGRAMME beliebig und mit relativ großer Dichte abgespeichert werden.
Zugriff und Datenübertragung erfolgen zwar ein wenig langsam, (ca.600 BITS/s) dagegen ist eine komfortablere Lösung aber auch wesentlich teuerer und anfälliger(Disketten). Im Hinblick auf eine leicht verständliche Schaltungstechnik wurde deshalb auch hier ein Kassetten-Interface zur Verfügung gestellt, daß preiswert und vor allem mit Sicherheit größere Mengen von Daten abspeichern kann, siehe BILD 11.
Um nun DATEN auf normale Tonband-Kassetten abspeichern zu können, müssen diese vorher in bestimmte Töne umgesetzt werden. Die MODULATOR-Schaltung des Kassetten-Interfaces hat deshalb die Aufgabe, den beiden LOGIK-PEGELN 0 u. 1 zwei definierte Frequenzen zuzuordnen. In unserem Fall bedient man sich eines sogenannten FSK-MODULATORS (Frequenz-Umtastung), der es ermöglicht mit den erzeugten TONFOLGEN 3.0 KHz und 6.0 KHz die digitalen Informationen vom System auf Band abzuspeichern.

Das von der CPU 8085 gelieferte, quarzstabile CLK-Signal von 3.072 MHz(6.144/2) wird von einem 14-stufigen Binärteiler C-MOS 14020 herabgeteilt. Am Ausgang 14 des Teilers steht dann eine Frequenz von 3.0 KHz zur Verfügung ($3.072\ MHz/2^{10} = 3.0\ KHz$), während am Ausgang 12 die Frequenz 6.0 KHZ ($3.072\ MHz/2^{9} = 6.0\ KHz$) beträgt.
Als Datenleitung zwischen dem CHIP-Computer und dem Kassetten-Interface wird die Leitung CO des KANALS C vom 8255A verwendet, siehe BILD 11. Abhängig vom Signalzustand dieser Leitung, gesteuert durch die SOFTWARE, gelangt eine der beiden Frequenzen über einen Spannungsteiler zum RECORDER-EINGANG. Die Frequenzumschaltung wird durch das FLIP-FLOP C-MOS 14013 zusammen mit den beiden NAND-GATTERN C-MOS 14011 so vorgenommen, daß das Ausgangssignal immer aus vollständigen PERIODEN des 3.0 KHz bzw. 6.0 KHZ-Signals besteht. Mit Hilfe eines Spannungsteilers kann anschließend die notwendige Pegelanpassung am Kassettenrecorder vorgenommen werden, diese Ausgangsspannung liegt in der Regel bei 400 mV.
Um empfangsseitig am Microcomputersystem den beiden TONFOLGEN wieder die entsprechenden Logikpegel "0" oder "1" zuordnen zu können, wird eine entsprechende DEMODULATOR-Schaltung nach BILD 11 verwendet. Zwei integrierte Schaltkreise LM 3900 u. 74LS04 bilden den sogenannten FSK-DEMODULATOR. Dabei arbeitet die erste Verstärkerstufe V1 als KOMPARATOR, um die vom Recorder gelieferten Eingangssignale in notwendige Rechtecksignale umzuwandeln. Der folgende Operationsverstärker V2 setzt entsprechend der Frequenz von 3.0 KHz oder 6.0 KHz seine Ausgangsspannung in kleine oder große Werte um. Über ein Tiefpaßfilter aufgebaut durch V3, gelangt das Signal zu einem zweiten KOMPARATOR V4, der wiederum die entsprechenden Logikpegel "0" oder "1" an seinem Ausgang zur Verfügung stellt. Die INVERTER-Schaltung durch 74LS04 erhöht anschließend die Flankensteilheit und ermöglicht eine extern anzuschließende optische Pegelanzeige.
Der USERRPORT BO des 8255A emfängt nun diese Daten und ein systemeigenes Steuerprogramm setzt diesen SERIELLEN Datenstrom in die entsprechenden 8-BIT-PARALLELEN Daten um.

Die Einstellung der DEMODULATOR-Schaltung über das 10 K - Potentiometer wird zweckmäßig über eine kleines Testprogramm vorgenommen, bei dem über den PORT CO ein symmetrisches Rechtecksignal aufgezeichnet wird. Das Impulsmuster wird dann so eingestellt, daß bei einem Eingangssignal aus 4 PERIODEN mit 3.0 KHz und 8 PERIODEN mit 6.0 KHz die Impulse für die Logikpegel "0" und "1" genau die gleiche Länge besitzen.
Da PORT B des EIN/AUSGABE-Bausteins 8255A im Recordermode auf EINGABE gelegt werden muß, ist es vorteilhaft diesen Anschluß schaltbar auszulegen. (siehe BILD 11)

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+       TESTPROGRAMM FÜR KASSETTEN-INTERFACE
+
+       ++++++++++++++++++++++++++++++++++++++++++
+            EDITIEREN EINES NEUEN FILES
+            NAME: TEST            DATUM: 1983
+       ++++++++++++++++++++++++++++++++++++++++++
+       NEW      ORG 8000H
+       NEW      MVI A,80H;Steuerwort 8255 alles Ausgabe
+       NEW      OUT 07H
+       NEW ADR1 MVI A,FFH
+       NEW      OUT 06H; CO auf log."1"
+       NEW      CALL 1E85H; sytemeigenes Zeitprogramm
+       NEW      SUB A; AKKU auf NULL
+       NEW      OUT 06H; CO auf log."0"
+       NEW      CALL 1E85H
+       NEW      JMP ADR1
+       NEW      END
+
+       STATUS:ASSEMBLE
+       STATUS:RUN FROM 8000H
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

2.3.4.1 DAS AUFZEICHNUNGSFORMAT

In BILD 11 ist die Übertragung eines Daten-Bytes von 23H dargestellt.

Jedes zu übertragende Zeichen besteht aus 8 Daten-Bit, 1 START-BIT und 2 STOP-BIT, also 11 BIT. Die Übertragung eines Bytes beginnt immer mit einem START-BIT von der Länge TO und endet mit einem STOP-BIT von 2xTO. Da bei diesem Aufzeichnungsverfahren nur über ein BYTE hinweg zwischen SENDER und EMPFÄNGER (Recorder) Synchronlauf vorhanden sein muß, spricht man auch von einer ASYNCHRONEN Datenübertragung. Weiterhin ist aus der Darstellung ersichtlich, daß das START-BIT als log."0" übertragen wird, es hat die Aufgabe beim späteren Einlesen der Daten vom Recorder eindeutig den Beginn der Datenübertragung eines Bytes anzuzeigen. Das STOP-BIT dient als Trennung zwischen den Daten-Bytes.
Mit Hilfe systemeigener Zeitunterprogramme (ADRESSE 1E85H) wird nun die Länge TO eines zu übertragenden DATEN-BITS festgelegt, dabei erfolgt die Datenübertragung mit einer Geschwindigkeit von 600 BITS/s = 600 BAUD. Diese Übertragungszeit wurde optimal an die DEMODULATOR-Schaltung des Interfaces angepaßt, um spätere Einlesefehler zu vermeiden.
Um nun mehrere PROGRAMME auf eine Kassette abspeichern zu können, fängt jede Datenübertragung mit einem definierten VORSPANN an, d.h einem Taktsignal der Frequenz, das eine Leuchtdiode zum BLINKEN veranlaßt. Durch diese optische Kennzeichnung wird jeder BEGINN einer Datenübertragung vom Recorder zum System eindeutig festgelegt.
Im einzelnen sieht diese SERIELLE Datenübertragung des CHIP-Computers folgendermaßen aus:
Zuerst wird die Endadresse, d.h. die letzte Zeile des QUELL-PROGRAMMS bestehend aus zwei BYTE übertragen. Jetzt folgen nacheinander die Inhalte der Speicherzellen, in denen das EDITIERTE Programm aufbewahrt ist. Nach beendeter Übertragung dieser Speicherinhalte wird ein 8-BIT-PRÜFSUMMEN-CODE erzeugt und ebenfalls mit übertragen. Beim Einlesen eines Programms wird durch kontinuierliches Abfragen jedes einzelnen BYTES durch KANAL BO die gesamte Information von der Kassette wieder aufgenaommen, START- u. STOP-BIT abgetrennt, und die Daten werden in PARALLELER Form wieder in die Speicherzellen abgelegt, in die sie ursprünglich vorhanden waren.

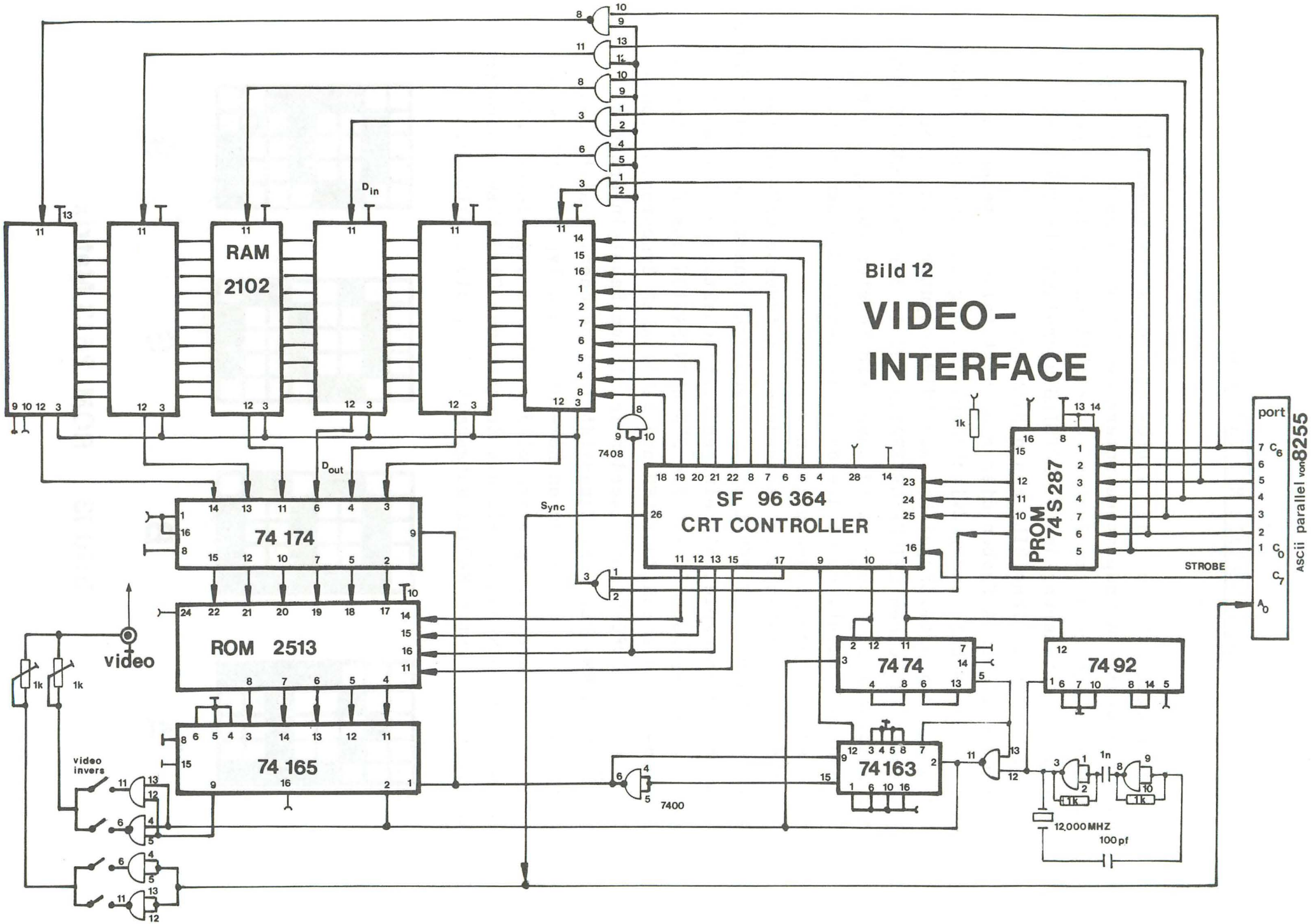


Bild 12
VIDEO-
INTERFACE
RAM
2102
D_{in}
D_{out}
74 174
ROM 2513
74 165
video
video invers
1k
7408
7400
Sync
SF 96 364
CRT CONTROLLER
PROM
74 S 287
STROBE
port
ascii parallel von 8255
74 74
74 92
74 163
12,000 MHZ
100 pf

2.3.5 DER VIDEOTEIL MIT CRT 96364A

Die komplette Schaltung des VIDEO-Interfaces ist ebenfalls auf der zweiten Grundplatine untergebracht und kann vorteilhaft unabhängig vom gesamten System auf Funktionstüchtigkeit geprüft werden. Die Steuerung des VIDEOTEILS übernimmt ebenfalls ein PORT des 8255A, siehe BILD 8 und BILD 12.
Die wesentlichen Aufgaben dieser Schnittstelle bestehen in dem Abruf und anschließender Zwischenspeicherung der darzustellenden Daten aus dem Arbeitsspeicher des CHIP-Computers, in ihrer Wiedergabe an einen ZEICHENGENERATOR und in der SYNCHRONISATION des Bildschirms. Dadurch wird quasi ein Fenster zum Speicher des Systems aufgemacht, wobei die darzustellenden Zeichen Reihe für Reihe auf den Schirm geschrieben werden.
Der für den CHIP-Computer verwendete, preiswerte CRT-Baustein 96364 von THOMSON CF führt alle notwendigen Funktionen selbständig aus, im Zusammenspiel mit einigen Hilfsschaltungen. Die wohl wichtigste Eigenschaft ist die SYNCHRONISATION des Bildschirms, das bedeutet die HORIZONTALEN und VERTIKALEN Synchronisationsimpulse des VIDEO-Signals werden in diesem Baustein erzeugt.
Der BILDSCHIRM-SPEICHER, Kapazität 1024 Zeichen, besteht aus einer Anzahl von 6 statischen RAM-Bausteinen Typ 2102 und wird während der Darstellung eines Bildes achtmal ausgelesen. Dies ist deshalb notwendig, weil die Zeichen des CHARAKTER-Generators in einer bestimmten MATRIX aufgebaut sind, siehe BILD 13.

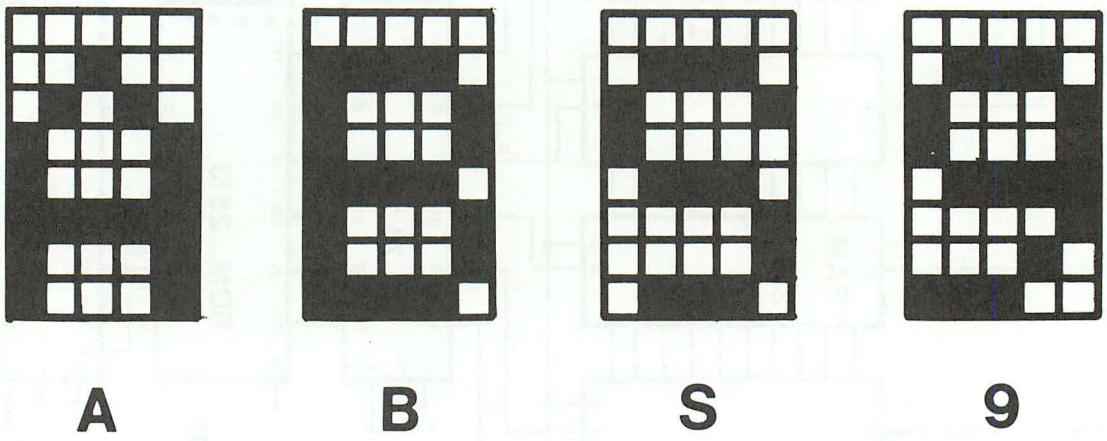


Bild 13 ROM 2513 Matrix

Der Zeichengenerator 2513 übersetzt das im Baustein 74LS174 zwischengespeicherte BIT-MUSTER der RAM 2102-Ausgänge, der anschließende PARALLEL-SERIELL-Wandlerbaustein 74165 führt die notwendige Umsetzung durch. Dieses Schieberegister wird mit einem sog. Bildpunkt-Signal gesteuert, dessen Frequenz bei 12.000 MHz liegt. Über Anschluß 10 des CRT-Controllers wird dieses Taktsignal noch synchronisiert, um alle acht REIHEN eines darzustellenden Zeichens untereinander stehen zu lassen. Mit diesem LINE-END-Signal, das nach dem 64. Zeichen auf log."0" geht wird die Zufuhr des CLOCK-Impulses bis zum Ende eines folgenden Synchronisationsimpulses blockiert.
Weiterhin sorgt der 96364A dafür, daß zwischen den aus 64 Zeichen bestehenden 16 Zeilen vier Zeilen des Bildrasters jeweils nicht moduliert werden. Die Höhe eines darzustellenden Zeichens belegt acht BILDRASTER-Zeilen. Mit dem 4-BIT-Binärzähler 74LS163, der ein Teilungsverhältnis von 1:8 besitzt, wird der Adresszähler (PIN 9) des CRT-Controllers gesteuert. Dies ist deshalb notwendig, um jedem BIT-MUSTER, das nur 5 BIT breit ist, einen Zwischenraum zuzuordnen, der zwischen den Zeichen in horizontaler Richtung durch das Schieberegister 74165 eingeschoben wird.
Mit Hilfe von diesem Schieberegister können also jedem Zeichen unmodulierte Punkte hinzugefügt werden, als Ergebnis ein Zwischenraum von 3 Punkten pro Zeichen, bei einer Gesamtbreite von 8 Punkten. Dieses PUNKTRASTER-Signal steht dann dem Ausgang auch invertiert zur Verfügung, um eine INVERSE Darstellung der Zeichen möglich zu machen.
Die beiden Potentiometer bestimmen das Amplitudenverhältnis zwischen VIDEO- u. SYNCHRONISATIONS-Signal, damit die Anpassung eines Monitors oder Fernsehgeräts leicht vorgenommen werden kann. (siehe auch 4.3.2)

2.3.5.1 DER ZEICHENGENERATOR ROM 2513

BILD 4.3.1 zeigt den CHARAKTER-Generator, einen ROM-Baustein mit einer Wortbreite von 5 BIT, wobei die Zeichenmatrix aus 5 SPALTEN und 8 ZEILEN besteht, die erste Zeile bleibt stets leer.

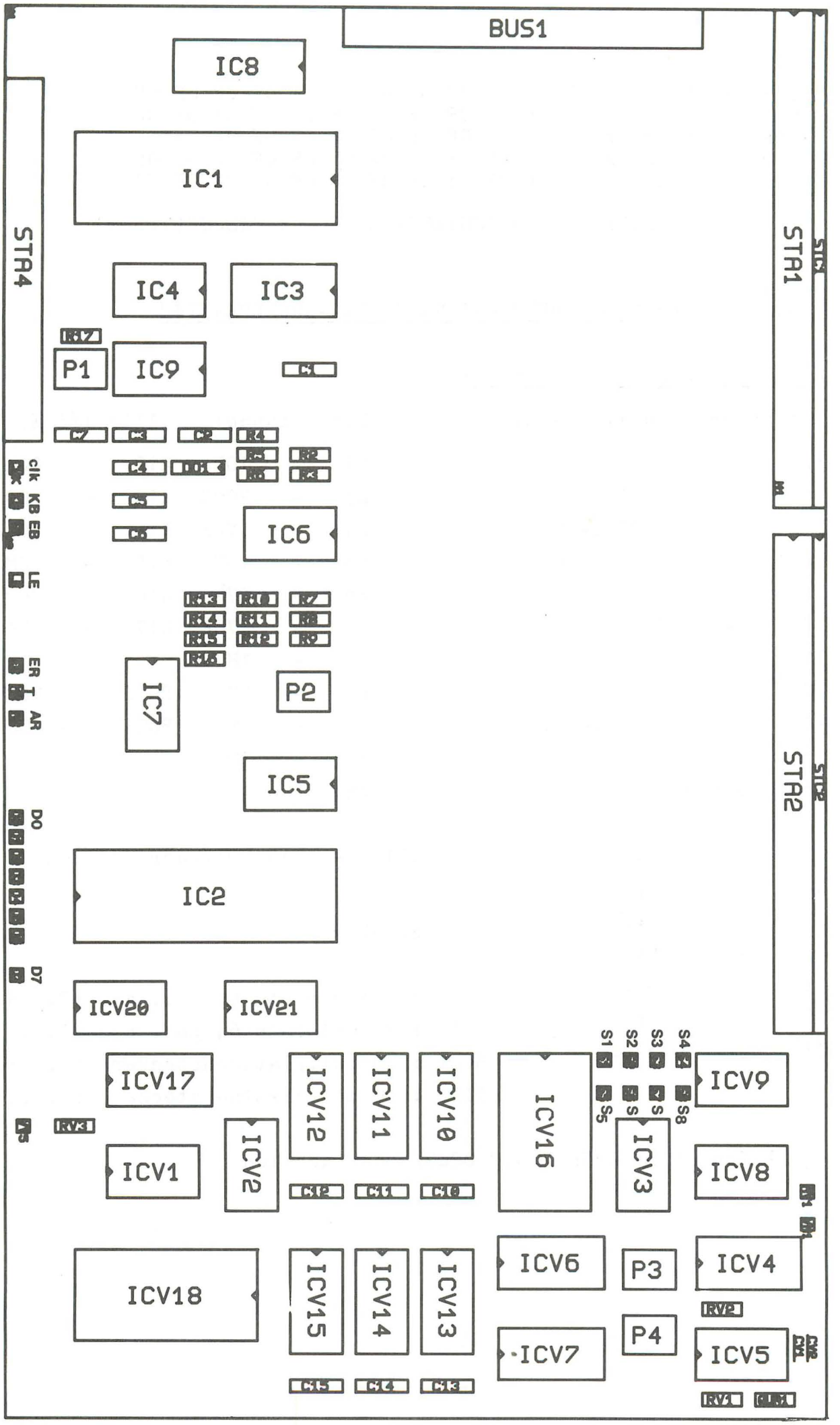
Dieser Baustein mit seiner 5 x 7 Punktmatrix ist verantwortlich für die Form, in der Buchstaben und Ziffern auf dem Bildschirm erscheinen. Reihenadresse und ASCII-Code bilden zusammen eine Adresse, durch die sich die verschiedenen Kombinationsmöglichkeiten ergeben. Das zur betreffenden Reihenadresse gehörende Bitmuster erscheint an den 5 Ausgängen des Bausteins, in Abhängigkeit vom anliegenden ASCII-CODE. Für jedes darzustellende Zeichen werden 8 Bitkombinationen benötigt, so daß 64 Zeichen zur Verfügung stehen.

2.3.5.2 PROGRAMMIERUNG DES CONTROL-DECODERS (74S287)

Der im BILD 12 dargestellte PROM-Baustein 74S287(74S387) liefert ein Ausgangssignal von drei BIT, das zu einem internen DECODER des CRT-Controllers gelangt.
Aus dem vom PORT C (C0-C7) kommenden ASCII-Signal, werden bestimmte Kontrollsignale abgeleitet, z.B. löschen einer Zeile oder des gesamten Bildes. Das Bitmuster an den Ausgängen dieses 256x4 BIT organisierten PROM-Bausteins bestimmt also wie der CRT-Controller 96 364 die angelegte Information auswertet.
Durch Programmierung diese Bausteins sind folgende Steuerfunktionen möglich:

CURSOR LINKS	08H	LÖSCHEN DES BILDES,CURSOR OBEN	0CH
CURSOR RECHTS	09H	RETURN UND LÖSCHEN BIS ZEILENENDE	0DH
CURSOR UNTEN	0AH	CURSOR OBEN LINKS (HOME)	1CH
CURSOR OBEN	0BH		

0000H	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08
0010H	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08
0020H	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08
0030H	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08
0040H	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08
0050H	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08
0060H	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08
0070H	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08	08
0080H	03	03	03	03	03	03	03	03	04	07	0A	06	08	09	03	03
0090H	03	03	03	03	03	03	03	03	03	03	0D	02	00	01	03	03
00A0H	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F


BUS1
IC8
IC1
STA4
STA1
IC4
IC3
P1
IC9
C1
C7
C3
C2
R4
R5
R2
C4
R6
R3
clk
KB
EB
C5
C6
IC6
LE
R13
R10
R7
R14
R11
R8
R15
R12
R9
R16
ER
AR
IC7
P2
IC5
STA2
D0
IC2
D7
ICV20
ICV21
S1
S2
S3
S4
ICV17
ICV12
ICV11
ICV10
ICV16
ICV9
S5
S
S
S8
RV3
ICV1
ICV2
ICV3
ICV8
C12
C11
C10
ICV6
P3
ICV4
ICV18
ICV15
ICV14
ICV13
RV2
P4
ICV7
ICV5
C15
C14
C13
RV1

00B0H	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F
00C0H	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F
00D0H	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F
00E0H	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F
00F0H	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	03

TABELLE 5 PROGAMMIERTABELLE FÜR 74S287(74S387)

2.3.7 BAUTEILE-ÜBERSICHT RAM/VIDEO-GRUNDPLATINE

2.3.7.1 Schnittstellenteil:

Integrierte Schaltungen:

IC 1	-	8255A
IC 2	-	8255A
IC 3	-	14020 C-MOS
IC 4	-	14013 C-MOS
IC 5	-	74LS00
IC 6	-	LM 3900
IC 7	-	74LS04
IC 8	-	74LS244
IC 9	-	14011 C-MOS

Widerstände: (Alle 1/4 W)

R2	-	10K	R12	-	22K
R3	-	390K	R13	-	100K
R4	-	470K	R14	-	100K
R5	-	10M	R15	-	10M
R6	-	47K	R16	-	200 OHM
R7	-	470K	R17	-	10K
R8	-	1M			
R9	-	270K	POTI P1	-	100K
R10	-	2M	POTI P2	-	10K
R11	-	1M			

Kondensatoren:

C1	-	22nF
C2	-	390pF
C3	-	330pF
C4	-	330pF
C5	-	330pF
C6	-	10nF
C7	-	0.33uF

Dioden:

DD1 - 1N4004/oder ähnlich

Stecker:

STA1 : Federleiste 64 pol. a+c (2.1.3)
STA2 : Federleiste 64 pol. a+c (2.1.3)
BUS1 : 40 pol.Platinenstecker (2.1.3)
STA4 : 40 pol.Platinenstecker (2.1.3)

ASCII-Tastatur: CHERRY G80-0247/ oder ähnlich

2.3.7.2 Videoteil:

Integrierte Schaltungen:

ICV 1	-	74LS08
ICV 2	-	74LS08
ICV 3	-	7400
ICV 4	-	74163
ICV 5	-	7400
ICV 6	-	74LS174
ICV 7	-	74165
ICV 8	-	7474
ICV 9	-	7492
ICV10	-	MM 2102 RAM
ICV11	-	MM 2102 RAM
ICV12	-	MM 2102 RAM
ICV13	-	MM 2102 RAM
ICV14	-	MM 2102 RAM
ICV15	-	MM 2102 RAM
ICV16	-	R03-2513 ROM (GENERAL INSTRUMENT)
ICV17	-	DM 74S287 PROM (NATIONAL SEMICOND.)
ICV18	-	EF 96 364 CRT-Controller (THOMSON CF)
ICV20	-	74LS125
ICV21	-	74LS125

Widerstände: (Alle 1/4 W)

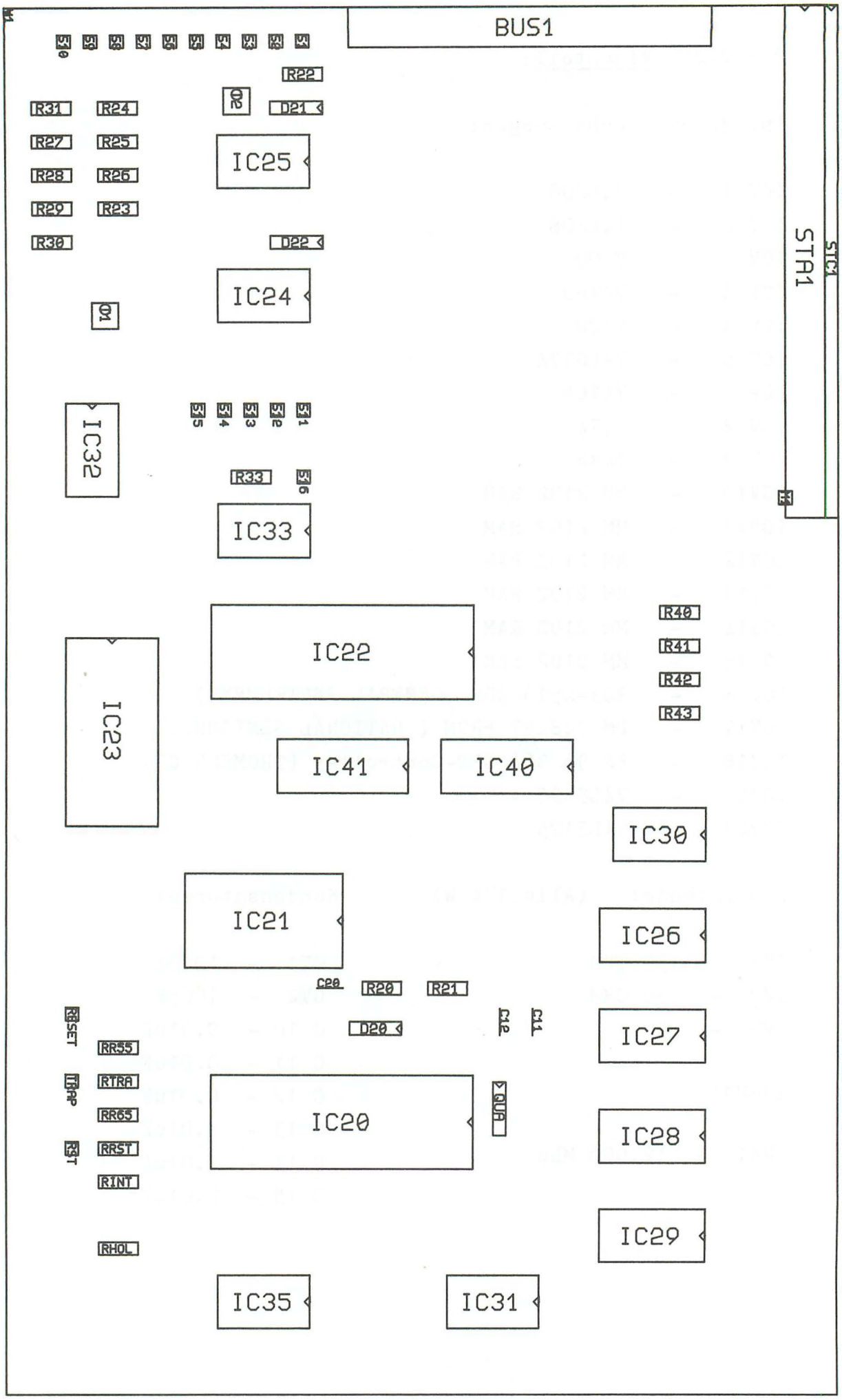
RV1	-	330 OHM
RV2	-	330 OHM
RV3	-	1K

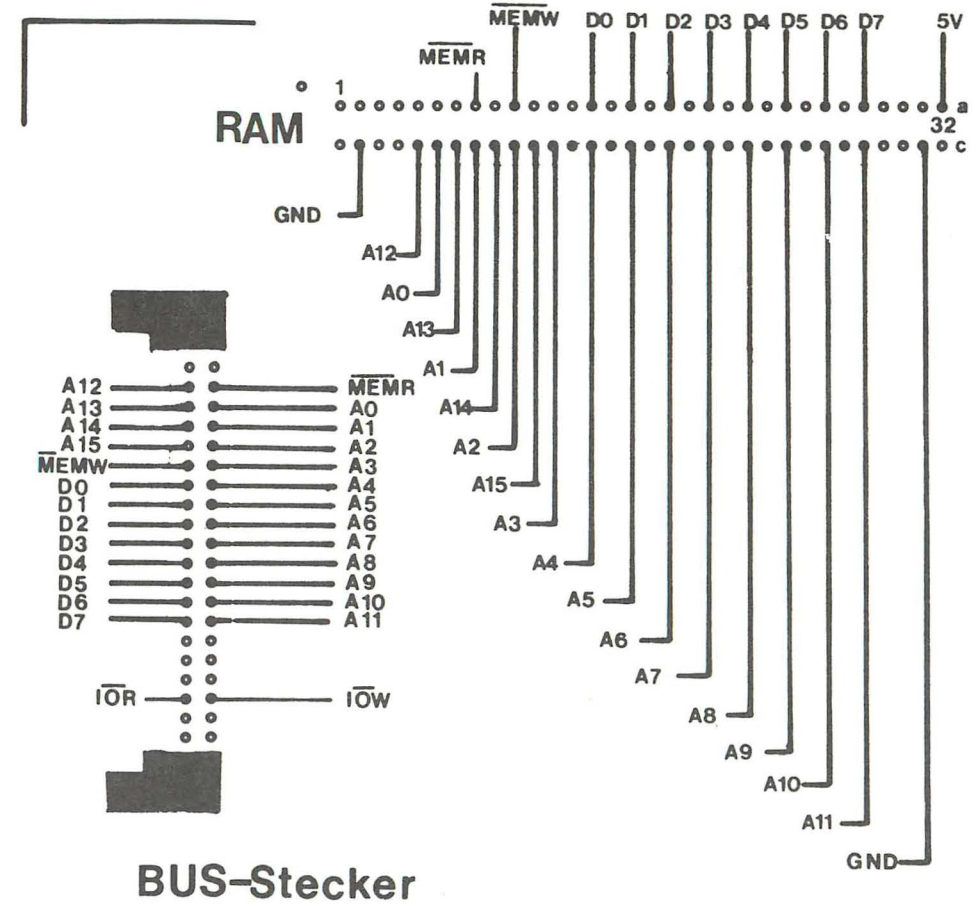
Quarz:

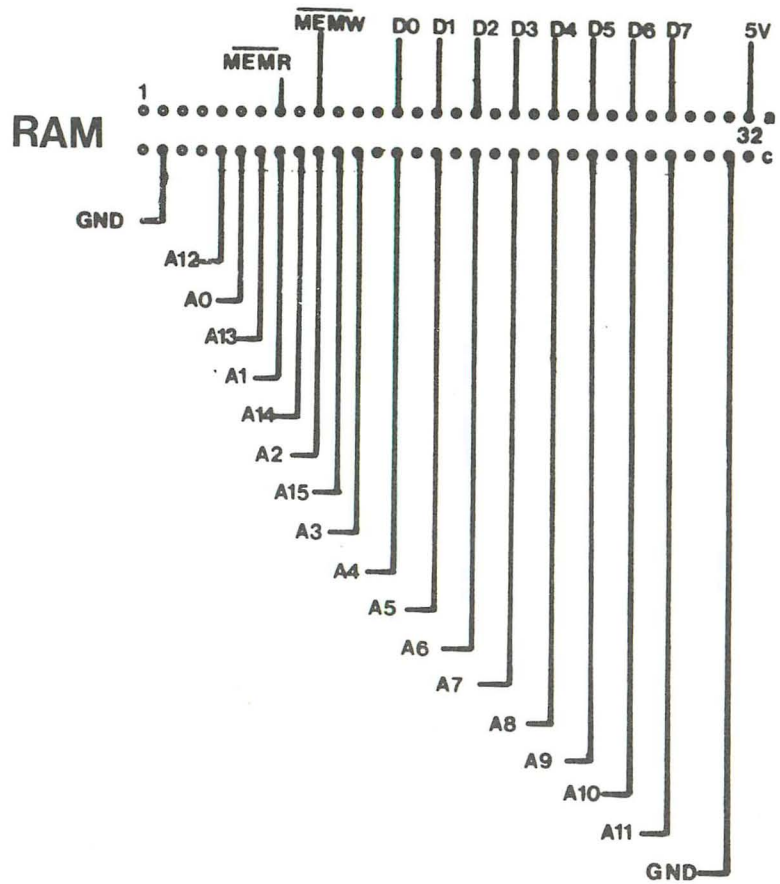
QUA1	-	12.000 MHz

Kondensatoren:

CV1	-	1000pF
CV2	-	100pF
C 10	-	0.01uF
C 11	-	0.01uF
C 12	-	0.01uF
C 13	-	0.01uF
C 14	-	0.01uF
C 15	-	0.01uF


BUS1
STA1
STC1
R22
D21
R31
R24
R27
R25
R28
R26
R29
R23
R30
D22
IC25
IC24
IC32
R33
IC33
IC22
IC23
R40
R41
R42
R43
IC41
IC40
IC30
IC21
IC26
C20
R20
R21
D20
C12
C11
IC27
RR55
RTRA
RR65
RRST
RINT
RHOL
IC20
QUA
IC28
IC29
IC35
IC31





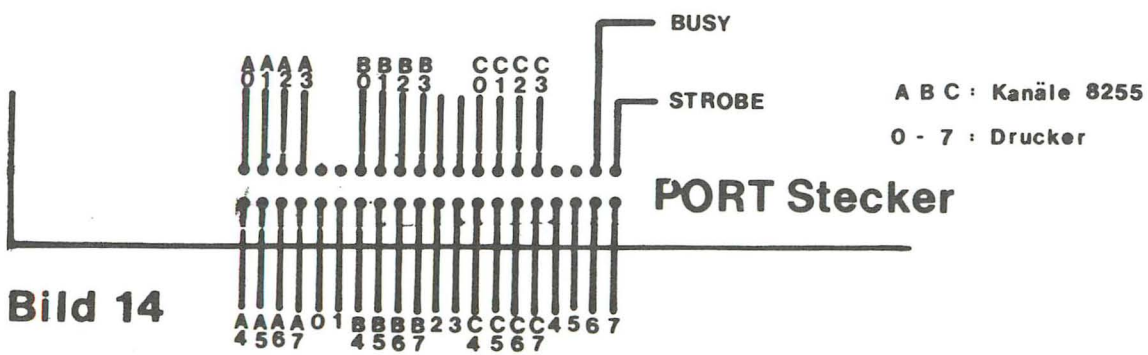


Bild 14

RAM VIDEO PLATINE

2.3.8 STECKERBELEGUNG RAM/VIDEO-GRUNDPLATINE

BILD 14 zeigt die Anschlußbelegung der beiden RAM-Speicherkarten über die Steckverbindung STA1 und STA2, sowie die BUS-Verbindung über BUS1 und PORT-Stecker STA4.
Laut Belegungsplan 2.3.6 haben folgende Anschlüsse die Bedeutung:

+ CLK Taktsignal für FSK von S 15 (siehe 2.1.4)
+ KB Schaltmöglichkeit von KANAL BO (siehe BILD 11)
+ EB "

+ ER Eingang Recorder , Ausgang des FSK-Modulators
+ MR MASSE
+ AR Ausgang Recorder , Eingang des FSK-Demodulators

+ LE Anschluß einer Leuchtdiode

+ DO BIT 0 Anschlüsse für ASCII-Tastatur
+ D1 BIT 1
+ D2 BIT 2
+ D3 BIT 3
+ D4 BIT 4
+ D5 BIT 5
+ D6 BIT 6
+ D7 STROBE

+ S4 + S8 VIDEO-Brücke: S1 + S6
+ S3 + S7 S4 + S8
+ S2 + S6
+ S1 + S5

+ AV1 VIDEO-Ausgang
+ VM1 MASSE

NETZTEIL

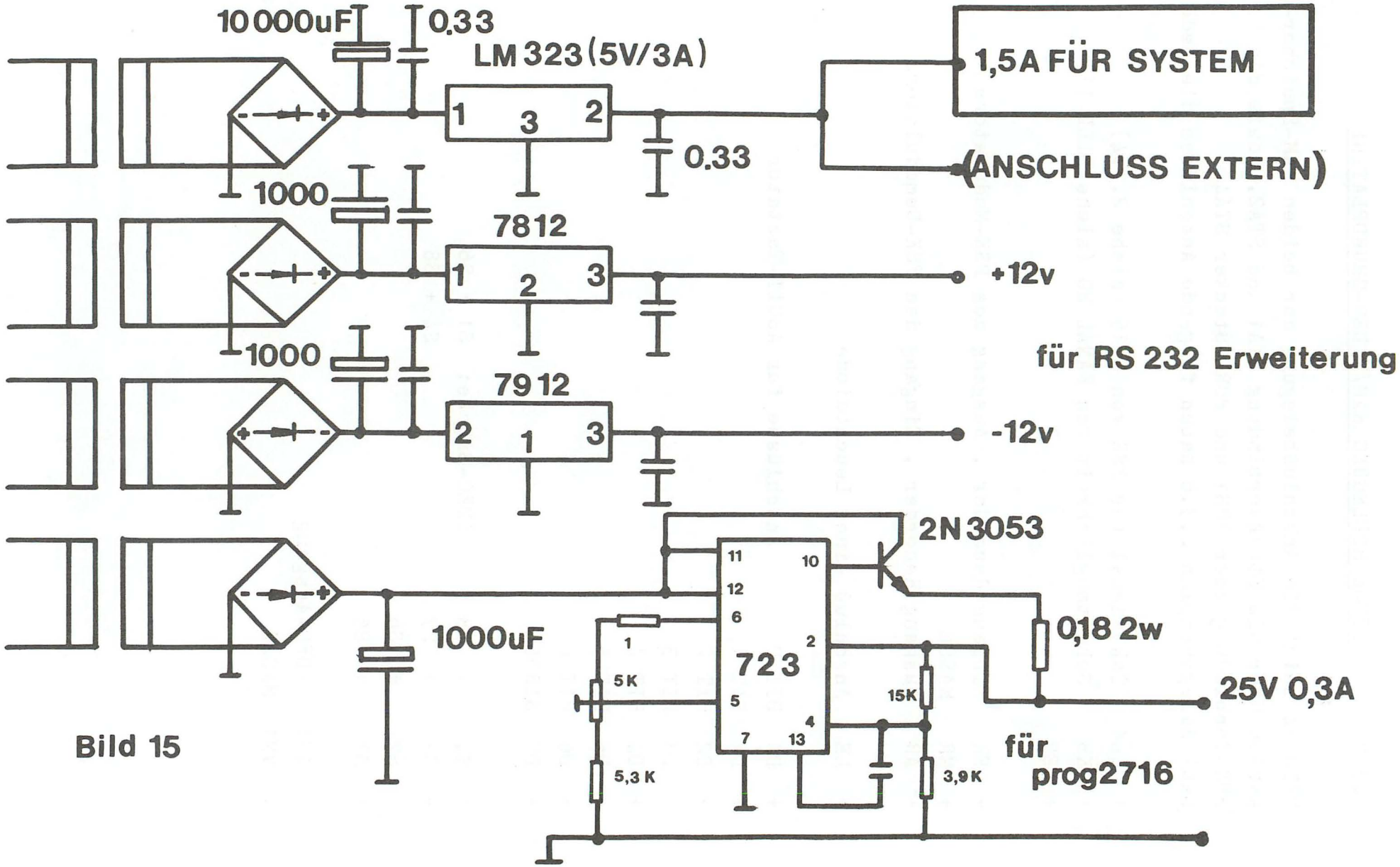


Bild 15

2.4 EIN NETZTEIL FÜR DEN CHIP-COMPUTER

BILD 15 zeigt einen Schaltungsvorschlag für die Spannungsversorgung des CHIP-Computers. Um für externe Schaltungen eine genügende Stromreserve zur Verfügung stellen zu können, wird die +5 V- Stabilisierung dort mit einem Spannungsregler LM 323 vorgenommen. Dieser Regler ist in der Lage, bei einer konstanten Spannung von +5 V, einen Strom von 3A zu liefern.
Die eigentliche Stromaufnahme des CHIP-Computers beträgt nur ca. 1,5A, man kann also ebenfalls über die bekannten, preiswerten Spannungsstabilisatoren des Typs 7805 mit Längstransistor die +5 V Versorgung vornehmen. Um dabei die Verlustleistung gering zu halten, sollte die unstabilisierte Eingangsspannung des mit einer Kühlfläche versehenden 5V-Reglers möglichst niedrig, bei ca. 8V liegen.
Für die serielle Schnittstelle des Systems, werden bei Bedarf die beiden Spannungsregler 7812 und 7912 eingesetzt. Die Bausteine dieser Serie eignen sich für einen preiswerten und problemlosen Aufbau stabilisierter Netzteile und haben einige deutliche Vorteile, wie interne Strombegrenzung, Kurzschlußfestigkeit und Schutz gegen Übertemperatur. An dieser Stelle sei auf die vielfältigen, praktischen Applikationen der Hersteller hingewiesen.
Die + 25 V- Programmierspannung des 2716-EPROMs wird mit Hilfe des bekannten Spannungsregler 723 erzeugt.
Zum Aufbau einer Stromversorgungseinheit läßt sich prinzipiell sagen, daß man nicht auf eine bestimmte Bauteilkonzeption für diese System angewiesen ist, doch sollte ein gebautes oder vorhandenes Netzteil in der Lage sein, die in BILD 15 dargestellten Spannungen und Ströme störungsfrei und sicher dem CHIP-Computer zu liefern.

2.4.1 ANSCHLUSS DER BETRIEBSSPANNUNG

Äußerst wichtig für den einwandfreien Betrieb des Systems ist die Belegung der auf den Bauteileplänen oder Platinenvorlagen fixierten

+5 V - u. MASSE-Anschlußpunkte, die alle und vor allem richtig angeschlossen werden müssen. Man muß unbedingt darauf achten, keine Verwechslung vorzunehmen, denn sonst könnte dies ev. zu einer Zerstörung bestimmter Bausteine führen.
Die Hilfestellung eines Strom-Meßgerätes zur ersten Inbetriebnahme wäre durchaus ratsam.
Die Aufteilung der +5 V- Versorgung auf mehrere Anschlußpunkte wurde im Hinblick auf den Störschutz vorgenommen, denn nicht ausreichende Maßnahmen auf diesem Gebiet können die Funktion des Gesamtaufbaus stark beeinträchtigen. Aus diesem Grund sind auch die Versorgungsspannungen der Speicher-und Schnittstellen-Platinen lt. Belegungsplan mit Kondensatoren gegen MASSE unbedingt abzublocken.

2.5 INBETRIEBNAHME DES CHIP-COMPUTERS

Vorausgesetzt wird für einen Funktionstest das einwandfreie Arbeiten eines in 2.4 beschriebenen und angeschlossenen Netzteils.
Da die Funktionsweise des gesamten Systems nur über den Bildschirm zu erkennen ist, sollte zu Beginn der VIDEO-Teil auf der zweiten Grundplatine getestet werden. Da die Funktionsfähigkeit dieser Schaltung unabhängig vom Gesamtsystem getestet werden kann, brauchen anfangs auch nur die beiden Anschlußpunkte zur Stromversorgung des VIDEO-Interfaces angeschlossen werden. Bei richtiger Arbeitsweise ergibt sich eine Stromaufnahme von ca. 700mA, wobei der gesamte Zeichensatz auf dem MONITOR oder FERNSEHER sichtbar wird. Bei der Verwendung eines üblichen Fernsehgeräts als Datensichtgerät müssen, je nach Herstellertyp, ev. gewisse Umbauten vorgenommen werden, um die erforderliche VIDEO-Bandbreite von mehr als 10MHz zu erzielen. Sinnvoll ist es in diesem Fall, den bekannten 5.5 MHz-Serienschwingkreis, die sog. Tonfalle aufzutrennen und das Signal direkt am Eingang des VIDEO-Transistors einzuspeisen. Über den VIDEO-Ausgang des Interfaces kann dann

mit Hilfe der beiden Potentiometer eine optimale Anpassung vorgenommen werden.
Arbeitet das VIDEO-Interface einwandfrei, so kann jetzt eine in 2.3.2 beschriebene ASCII-Tastatur einschl. BREAK(RESET)-Möglichkeit angeschlossen werden.
Sämtliche Speicherkarten EPROM u. RAM werden nun in die im System vorhandenen STECKER angeschlossen, ebenso die restlichen Stromversorgungen. Man muß darauf achten, daß auf den drei Speicherkarten die Einstellung der ADRESSEN-BLÖCKE in der richtigen Weise vorgenommen worden ist. Auf der 16K-Byte-EPROM-Platine ist die Reihenfolge der Bausteine EPROM 1 bis EPROM 8 von entscheidender Bedeutung, eine Verwechslung würde bedeuten, daß das BETRIEBSSYSTEM überhaupt nicht oder nur teilweise funktioniert.
Wie eingangs schon erwähnt, müssen alle Platinen des CHIP-Computers mit der notwendigen Sorgfalt hergestellt u. bestückt werden. Ist dies noch einmal kontrolliert worden, so kann durch die Verbindung mit dem 40 pol. BUS-Kabel der hardwaremäßige Aufbau des Systems abgeschlossen werden.
Nach dem Einschalten der Versorgungsspannung meldet sich der CHIP-Computer mit einer sog. STATUS: Linie am untersten Rand des Bildschirms über die nun die gesamte Befehlseingabe durch die Tastatur läuft.
An dieser Stelle seien dem Anwender noch folgende,erfahrungsmäßige Hinweise zur ev. Fehlersuche gegeben:

1. Spannungsversorgung an allen vorgeschriebenen Punkten(4,8-5.0V)?
2. BUS-kabel richtig angeschlossen?
3. Alle Bauteile auf Grundplatinen u. Karten richtig bestückt?
4. Adressierung der RAM-Karten auf 4000H u. 8000H (S1u.S2)?
5. Arbeitet EPROM-Karte im ROM-Mode (S5+S6)?
6. Alle ADRESS,DATEN u. STEUER-Leitungen an den RAM-Karten?
7. Sämtliche $\overline{CS}$-Anschlüsse vorhanden?
8. Überprüfung der Leiterbahnführung, keinen Kurzschluß erzeugt?
9. ASCII-Tastatur BIT1-BIT7 u. STROBE richtig angeschlossen?
10. EPROM-Bausteine in der vorgeschriebenen Reihenfolge?

2.6 ERWEITERUNG DER HARDWARE DES CHIP-COMPUTERS

Wie man später bei der Programmerstellung sieht, wird die Funktionsweise des CHIP-Computers weitgehend durch die SOFTWARE bestimmt. Die HARDWARE hingegen hat nämlich zu gewährleisten, daß der Durchgriff vom PROGRAMM her nicht nur auf den Microcomputer selbst beschränkt bleibt, sondern auch noch die angeschlossenen SCHNITTSTELLEN erfaßt.

2.6.1 ANSCHLUSS EINER LEUCHTDIODEN-REIHE

Mit der in BILD 11 dargestellten Schaltung besteht nun die Möglichkeit, eine 8-stellige LED-Reihe an den gepufferten Ausgang des USERPORTS A anzuschließen. Damit ergeben sich dann später einige interessante Programmierübungen, die das Verständnis des vorhandenen Schnittstellen-Bausteins 8255A vertiefen sollen. BILD 16 zeigt diesen Anschluß.

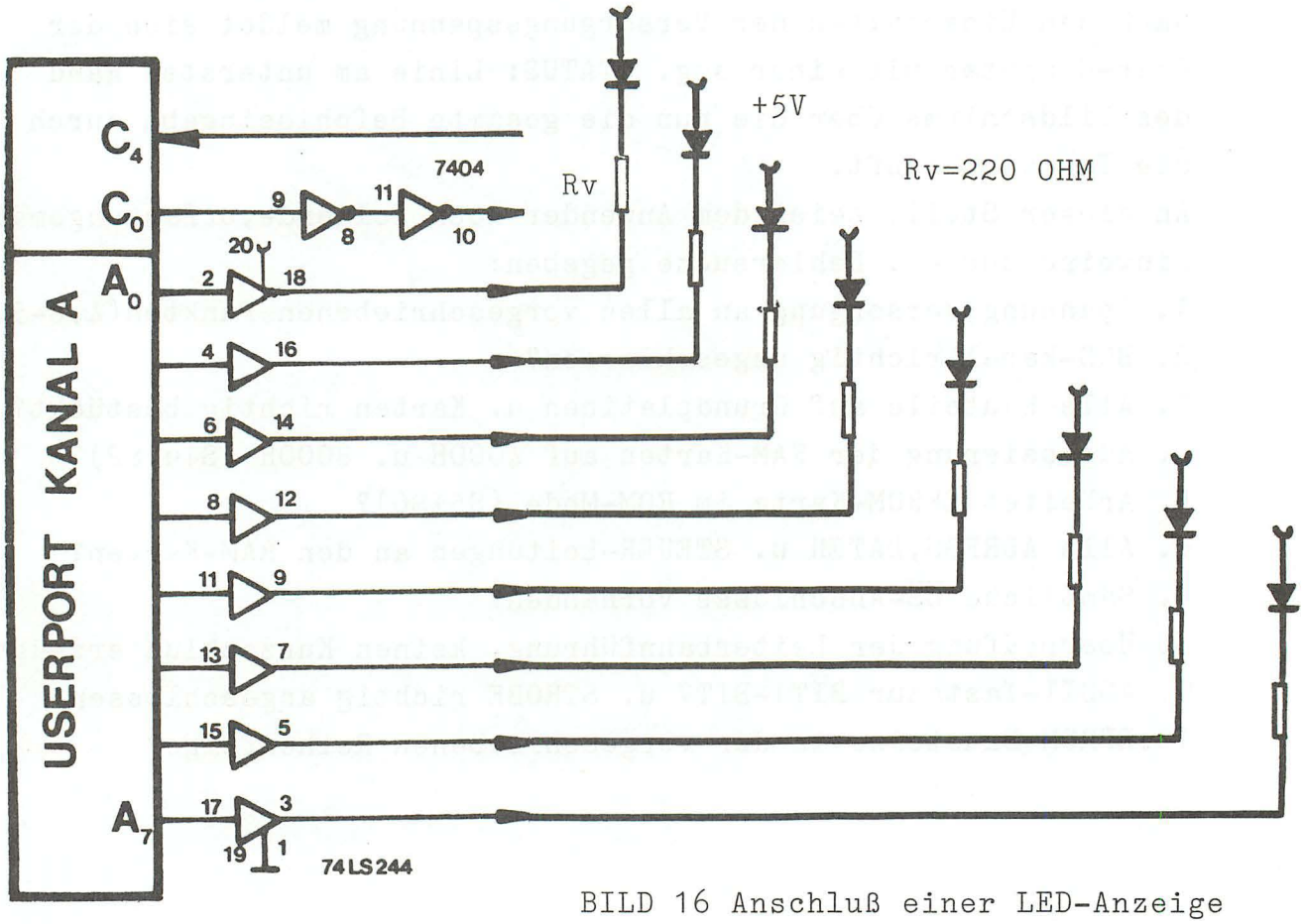


BILD 16 Anschluß einer LED-Anzeige

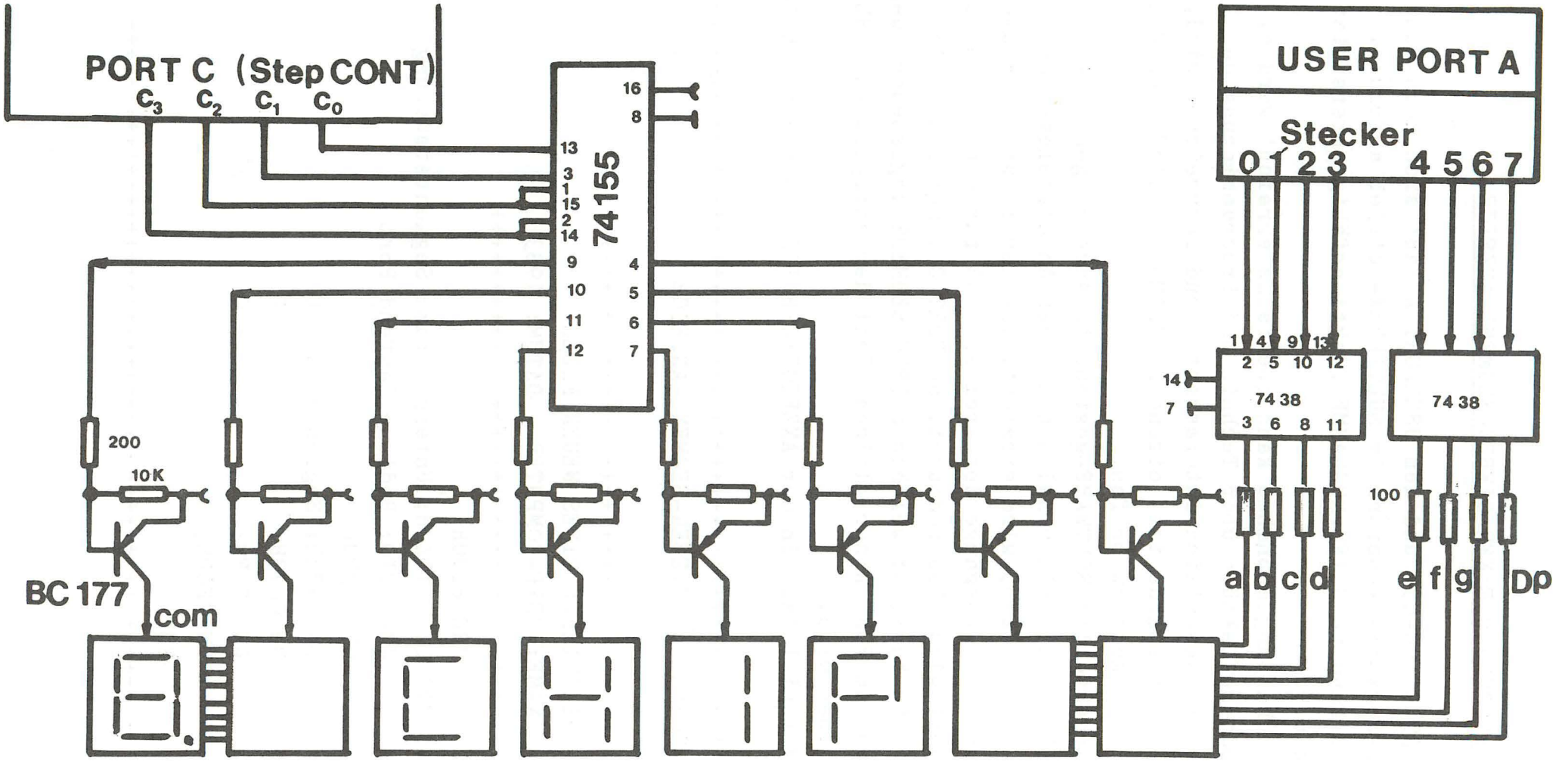


Bild 17 **Siebensegment-Steuerung**

2.6.2 ANSCHLUSS EINER SIEBEN-SEGMENT-ANZEIGE

Über den PORT C (CO-C3) und dem USERPORT A läßt sich nach BILD 17 eine Sieben-Segment-Steuerung im MULTIPLEX-Betrieb aufbauen, die durch vorhandene UNTERPROGRAMME funktionsfähig unterstützt wird, siehe 6.9.
Mit dieser Schaltungsmaßnahme kann ebenfalls erreicht werden, daß eine spätere Lektüre über Technik und Programmierung des Microprozessors wesentlich interessanter und lehrreicher wird, wenn der Leser imstande ist, eigene PROGRAMME in einfacher Weise auf ihre Richtigkeit zu prüfen.
Die hier dargestellte SOFTWARE-gesteuerte Anzeige gestattet die Darstellung beliebiger Zeichen auf den Anzeigeelementen.
Über den Baustein 74155 wird erreicht, daß immer nur ein Element der Reihe über die Ausgänge der Treiber 7438 für eine kurze Zeit durchgesteuert wird. Das bedeutet, am PORT C liegt immer die Adresse an, die bestimmt welches SIEBEN-SEGMENT angesteuert werden soll, während am PORT A zur gleichen Zeit der entsprechende SEGMENT-CODE ausgegeben wird.
Folgendes Beispiel läßt in der ANZEIGE:C H I P 8 0 8 5 erscheinen:

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+       PROGRAMM FÜR SIEBEN-SEGMENT-ANZEIGE
+
+       ++++++++++++++++++++++++++++++++++++++
+            EDITIEREN EINES NEUEN FILES
+            NAME: CHIP-COMPUTER   DATUM: 1983
+       ++++++++++++++++++++++++++++++++++++++
+       NEW      ORG 8000H
+       NEW OTTO CALL 1D43H; Unterprogramm Segmentsteuerung
+       NEW      JMP OTTO; Bediene laufend Port
+       NEW      ORG 83F8H
+       NEW      DW 3976H; Segmentcode C H
+       NEW      DW 0673H;             I P
+       NEW      DW 7F3FH;             8 0
+       NEW      DW 7F6DH;             8 5
+       NEW      END
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

2.6.3 EINE EPROM-PROGRAMMIEREINRICHTUNG

BILD 18 zeigt die Möglichkeit einer EPROM 2716 Programmier-Einrichtung, die ebenfalls bereits softwaremäßig im Betriebssystem des CHIP-Computers vorhanden ist. Über diese Programme wird dem Anwender die Möglichkeit gegeben, an einem Festwertspeicher diesen Typs folgende Funktionen auszuführen:

1. DATEN zu programmieren
2. DATEN des EPROMS zu lesen
3. DATEN zu vergleichen (verifizieren)

Alle o.a. Funktionen lassen sich über die Anschlüsse der USERPORTS des 8255A durchführen, wobei im PROGRAMMIER-MODE noch folgendes näher erläutert werden muß, siehe 3.3 :
Im Ursprungszustand, sowie nach jedem Löschvorgang führen alle BITS des 2716 den Zustand log."1". Die DATEN werden durch selektives Programmieren von log."0" in die gewünschten BIT-Plätze eingegeben. Die einzige Möglichkeit eine log."0" wieder in eine log."1" umzuwandeln, ist die LÖSCHUNG durch UV-Licht.
Das EPROM 2716 befindet sich in der Betriebsart : PROGRAMMIEREN, wenn am Anschluß Vpp die Versorgungsspannung +25V beträgt und $\overline{OE}$ auf log."1" liegt. Die zu programmierenden DATEN werden 8-BIT-PARALLEL an die Daten-Ausgabe-Anschlüsse angelegt. Wenn die Adresse und die Daten stabil sind, wird ein aktiver log."1" Programmier-Impuls von 50ms an den Eingang $\overline{CE}$ gelegt. Für jede zu programmierende Adresse muß ein Programmier-Impuls angelegt werden, die Länge dieses Impulses darf aber den Wert von 55ms nicht überschreiten.

2.6.4 DIE SERIELLE SCHNITTSTELLE 8251A

Der in BILD 2 dargestellte SCHNITTSTELLEN-Baustein 8251A hat die Aufgabe im Zusammenspiel mit einigen Hilfsschaltungen als SERIELLER

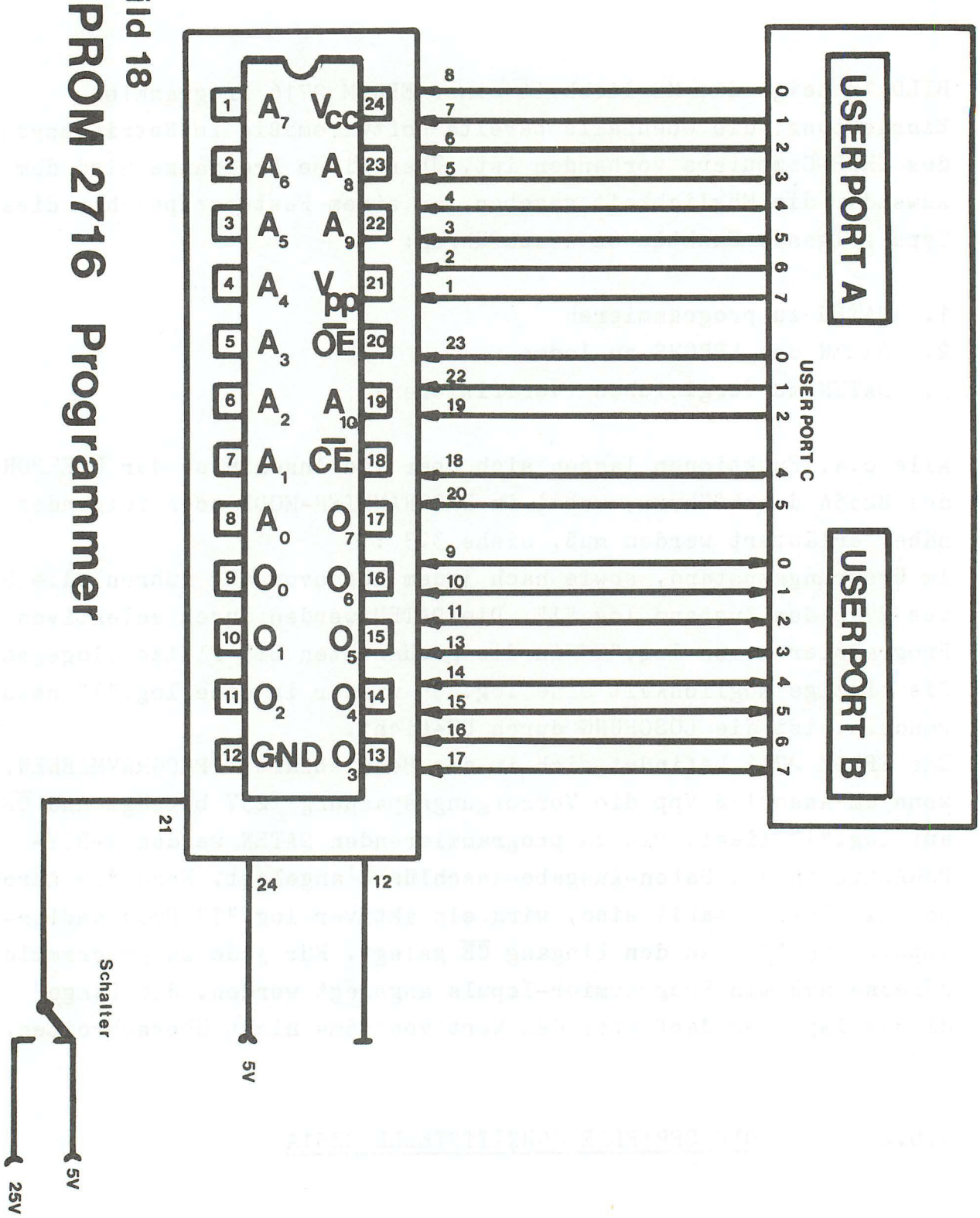


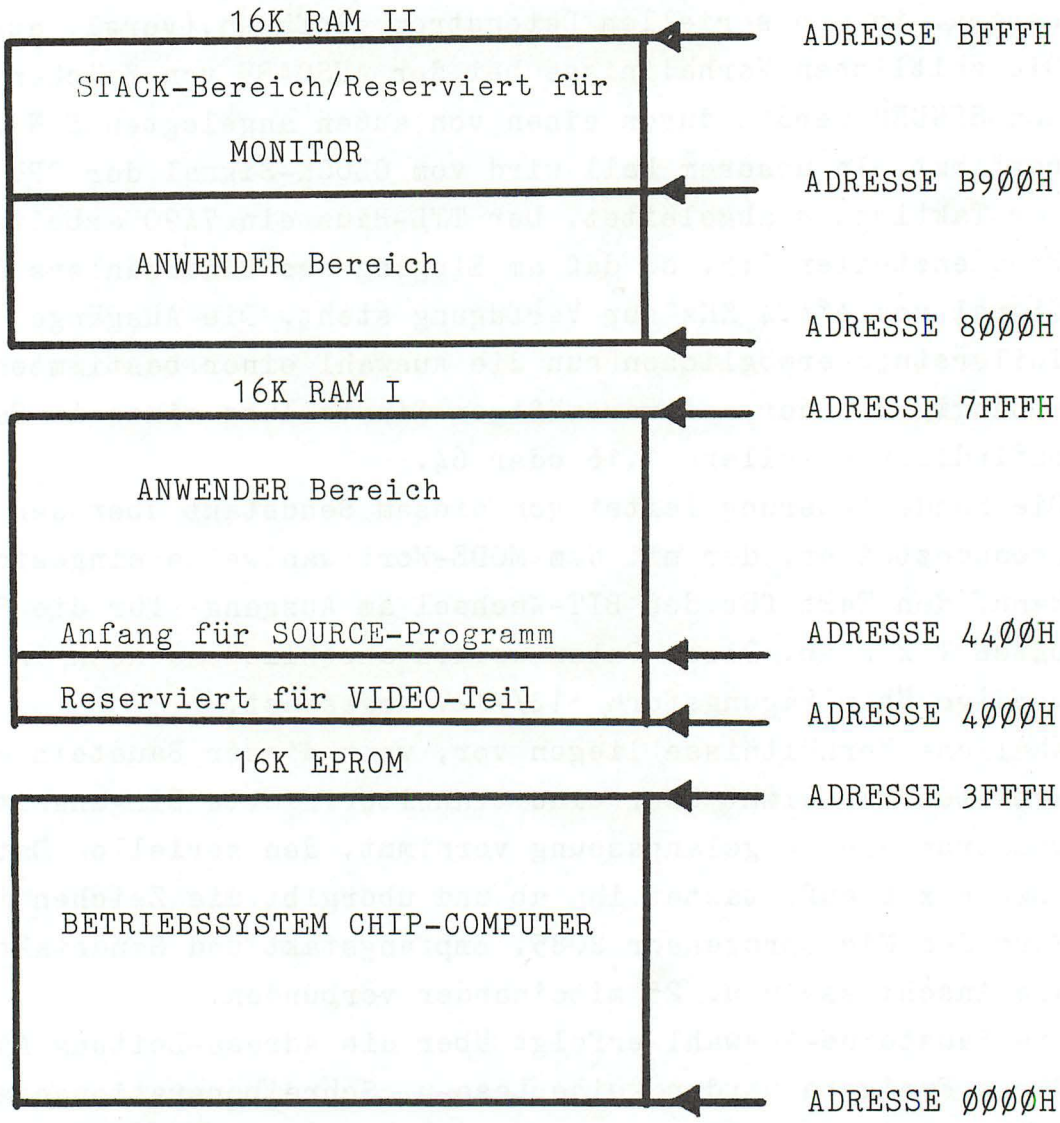
Bild 18

EPROM 2716 Programmer

Sender und Empfänger zu arbeiten. Er übernimmt vom Microprozessor 8085 Zeichen in Parallel-Darstellung und wandelt sie für das Senden in einen seriellen Datenstrom um. Andererseits kann er einen seriellen Datenstrom empfangen und daraus die für die CPU notwendige Parallel-Darstellung erzeugen.
Der 8251 ist in der Lage asynchrone als auch synchrone Daten zu übertragen. Bei der asynchronen Datenübertragung wird jedes zu übertragende Zeichen von einem STARTBIT und von 1,1,5 oder 2 STOP-BIT eingerahmt. Der Sender muß also außer der benötigten Wandlung noch BITS oder Zeichen, die für eine Übertragungsform benötigt werden, in den seriellen Datenstrom einfügen.(vergl. auch mit 2.3.4).
Die zeitlichen Verhältnisse bei der AUSGABE von Zeichen durch den SENDER werden durch einen von außen angelegten SENDE-TAKT bestimmt. In unserem Fall wird vom CLOCK-Signal der CPU 8085(3.072MHz) der Taktimpuls abgeleitet. Der TTL-Baustein 7490 arbeitet als Frequenzteiler 1:5, so daß am Eingang des Binärzählers 7493 ein Signal von 614.4 KHz zur Verfügung steht. Die Ausgänge dieser Teilerstufe ermöglichen nun die Auswahl einer bestimmten BAUD-Rate, abhängig von der softwaremäßigen Einstellung eines im Baustein 8251A befindlichen Teilers 1,16 oder 64.
Die Sendesteuerung leitet von diesem Sendetakt über den internen Frequenzteiler, der mit dem MODE-Wort wahlweise eingestellt werden kann, den Takt für den BIT-Wechsel am Ausgang für die SERIELLEN Daten T x D ab. Diese Daten werden anschließend noch in die notwendige Übertragungsform +12/-12V umgesetzt.
Ähnliche Verhältnisse liegen vor, wenn dieser Baustein als EMPFÄNGER arbeitet. Er nimmt über eine transitorisierte Eingangsstufe, die wiederum eine Pegelanpassung vornimmt, den seriellen Datenstrom über R x D auf, tastet ihn ab und übergibt die Zeichen in paralleler Form dem Microprozessor 8085. Empfangstakt und Sendetakt sind über die Anschlüsse 9 u. 25 miteinander verbunden.
Die Bausteine-Auswahl erfolgt über die Adress-Leitung A5, ohne diese Freigabe werden keine Lese-u. Schreiboperationen ausgeführt.
BILD 4.1.4 zeigt den internen Aufbau dieses SCHNITTSTELLEN-Bausteins.
Eingehendere Informationen, insbesonders zur PROGRAMMIERUNG können

durch umfangreiche Beschreibungen erworben werden, an dieser Stelle ist es nicht möglich die Vielfältigkeit dieses Bausteins auf einige Seiten begrenzt in sinnvoller Form aufzuzeigen (siehe Literaturhinweise).

2.7 SPEICHEREINTEILUNG CHIP-COMPUTER



3.1 Der Mikroprozessor SAB 8085 A

Der SAB 8085 A, aus dem Familienkonzept der 8080 - 8085 Intelfamilie, ist ein 8-Bit-Prozessorbaustein, d. h. er verarbeitet 8 Bit parallel.

Der SAB 8085 A enthält, zum Unterschied zu seinem weltweit verbreiteten Vorgänger SAB 8080 A, auch noch die Takterzeugung, die System-Bus-Steuerung und die Prioritätsauswahl für die Unterbrechungssteuerung. Weiters unterscheidet er sich von seinem Vorgänger durch:

- eine um ca. 50% herabgesetzte Zykluszeit
- vier Unterbrechungseingänge (Interrupts, einer nicht maskierbar) und eine zusätzliche, mit dem SAB 8080 A kompatible Unterbrechungssteuerung
- Kanäle für serielle Dateneingabe/Datenausgabe
- nur eine Versorgungsspannung (5 V).

Der Befehlssatz des SAB 8085 A ist auf Assembler- und Maschinenebene aufwärts kompatibel zum System 8080.

Der Adress-Bus enthält 16 Adressleitungen. Dies bedeutet, daß 2^{16} = 65 536 Bytes (= 64 KByte) adressiert werden können.

Weiters erzeugt der SAB 8085 A Steuersignale, die zur Auswahl externer Bausteine und zur Durchführung von Lese- und Schreiboperationen verwendet werden können, sowie zur Auswahl von Speichern und Ein-/Ausgabekanälen. Der SAB 8085 A kann bis zu 256 verschiedene E/A-Kanäle adressieren.

3.1.1 Register

Register sind kleine, schnelle Zwischenspeicher (RAMs), die normalerweise gerade 1 Byte fassen.

Der SAB 8085 A ist wie der SAB 8080 A mit internen 8-Bit-Registern und 16-Bit Registern ausgestattet. Er hat 8 adressierbare 8-Bit-Register. 6 davon können entweder als 8-Bit-Register oder als 16-Bit-Registerpaare verwendet werden. Registerpaare

werden so behandelt, als wären sie einzelne 16-Bit-Register; das höherwertige Byte eines Paares ist in dem ersten Register untergebracht, das niederwertige Byte in dem zweiten. Zusätzlich zu den Registerpaaren enthält der SAB 8085 A noch 2 weitere 16-Bit-Register.

Die Register des SAB 8085 A werden wie folgt unterschieden:

- Akkumulator (ACC oder Register A)

  Der Akkumulator ist ein 1-Byte-Register, das u. a. direkt von außen geladen werden kann. Es enthält stets das Ergebnis der gerade durchgeführten ALU-Operationen.

  Zu den Akkumulator-Befehlen gehören arithmetische, logische, Lade- und Speicherbefehle sowie E/A-Anweisungen.

- Befehlszähler (Program counter, PC)

  Der Befehlszähler zeigt jeweils auf den Speicherplatz des nächsten auszuführenden Befehls. Er enthält stets eine 16-Bit-Adresse und ist von außen nicht zu laden. Nullsetzen erfolgt durch RESET.

- Mehrzweckregister

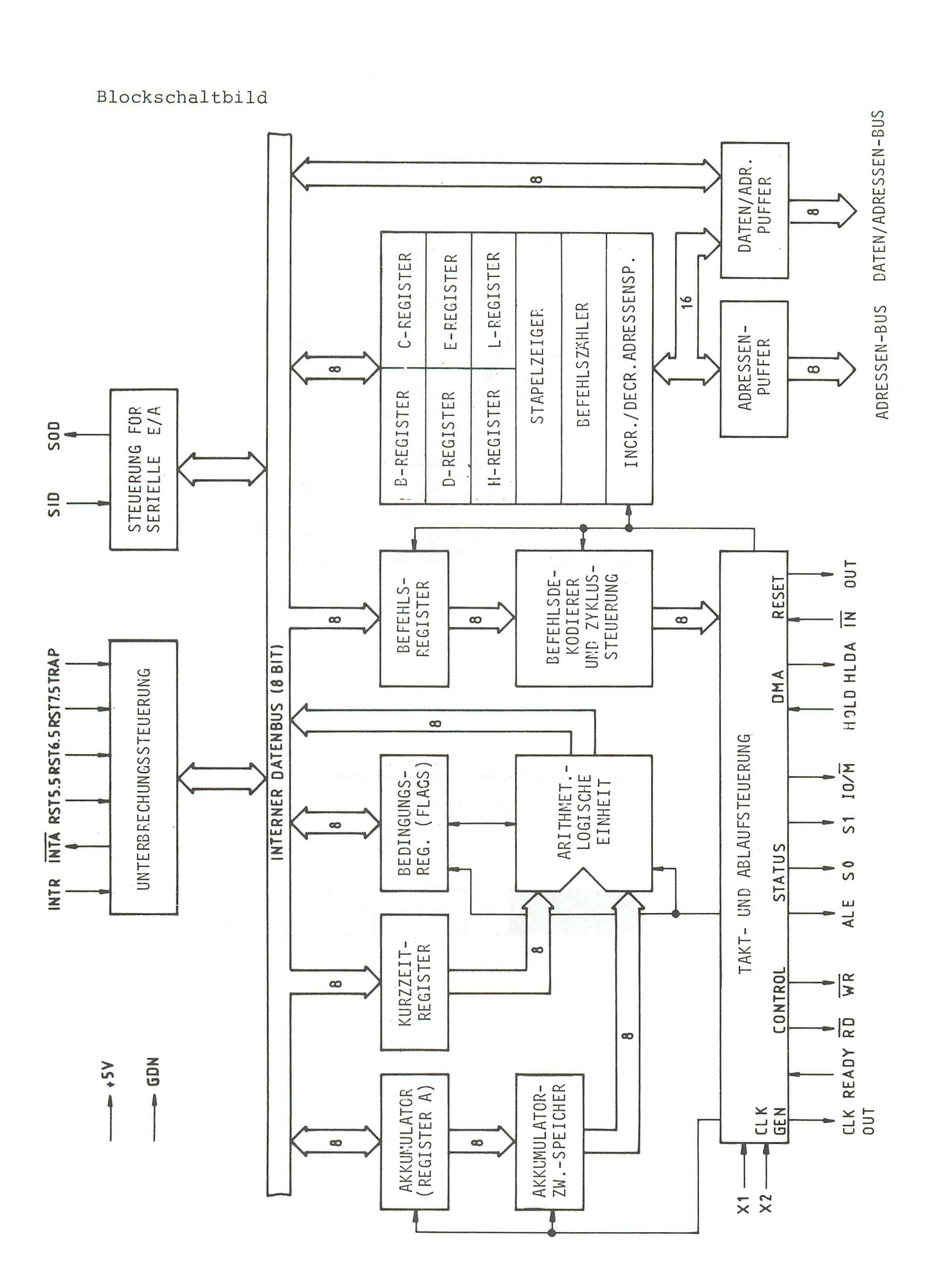
  Die Mehrzweckregister BC, DE und HL können als 6 Register zu je 8 Bits oder als 3 Register zu 16 Bits verwendet werden, abhängig vom auszuführenden Befehl. Sie stehen mit dem internen Daten-Bus mit der CPU in Verbindung.

  Das Register HL funktioniert als Datenzeiger zum Ansprechen von Speicheradressen, welche bei einer Anzahl von Befehlen entweder die Quellen oder die Ablegeplätze von Daten sind. Bei einer kleinen Anzahl von Anweisungen können auch die Register BC oder DE für die indirekte Adressierung verwendet werden.

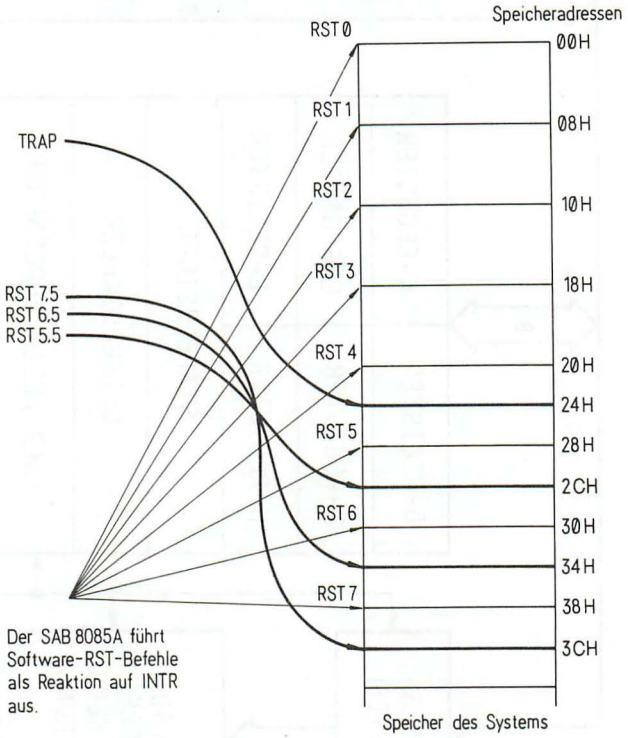
- Stapelzeiger (Stackpointer, SP)

  Der Stapel-Speicherbereich dient im äußeren RAM-Bereich für die zeitweilige Speicherung von Daten und Adressen. Der Stapelzeiger ist ein besonderer Datenzeiger, der stets auf die

Blockschaltbild



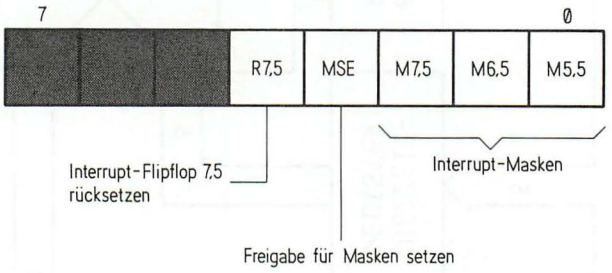
Lage der Hard- und Software-RST-Verzweigung



Setzen der Interrupt-Masken durch den Befehl SIM

SIM-Interrupt-Maske setzen
(Operationscode = 30)

Akkumulatorinhalt vor Ausführung von SIM



Adresse zeigt, die die letzte gültige Eintragung enthält. Der Stapelzeiger ist ein unteilbares 16-Bit-Register.

- Bedingungsregister (Flag-Register)

  Das Bedingungsregister enthält 5 Flags von je 1 Bit, in welchen je eine Zustandsinformation des Prozessors registriert wird; dadurch kann die Arbeitsweise des Prozessors gesteuert werden.

  Das Bedingungsregister wird nur durch arithmetische und logische Befehle gezielt beeinflußt.

Flags

Nachstehend sind die 5 Flags in der Zentraleinheit des SAB 8085 A dargestellt:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	$D_Ø$
S	Z		AC		P		CY

Das Carry-Flag (CY) (Übertrags-Flag) wird durch arithmetische Operationen gesetzt und rückgesetzt. Sein Zustand kann direkt per Programm abgefragt werden. Beispielsweise kann die Addition von zwei 1-Byte-Zahlen eine Summe ergeben, welche nicht in einem Byte untergebracht werden kann.

Hexadezimal	Binär
AEH	1 Ø 1 Ø 1 1 1 Ø
+ 74H	Ø 1 1 1 Ø 1 Ø Ø
122H	$\boxed{1}$ Ø Ø 1 Ø Ø Ø 1 Ø

Das Übertrags-Bit setzt das Carry-Flag auf 1

Ein Additionsvorgang, der zu einem Überlauf aus dem höchstwertigen Bit des Akkumulators führt, setzt das Carry-Flag. Andererseits wird das Carry-Flag durch einen Additionsvorgang,

der zu keinem Überlauf führt, gelöscht. Das Carry-Flag fungiert auch das "Borge"-Flag bei Subtraktionsvorgängen.

Das Hilfscarry-Flag (AC) zeigt einen Überlauf aus dem Bit 3 des Akkumulators in der gleichen Weise, wie das Carry-Flag einen Überlauf aus dem Bit 7 anzeigt. Dieses Flag wird im allgemeinen für die BCD-Arithmetik (Arithmetik mit binär codierten Dezimalzahlen) verwendet.

Das Vorzeichen-Flag (S) wird entsprechend dem Zustand des hächstwertigen Bits im Akkumulator nach Durchführung von arithmetischen oder logischen Befehlen gesetzt. Bei diesen Befehlen können 7 Datenbits zur Darstellung des im Akkumulator stehenden Zeichens oder der entsprechenden Zahl verwendet werden. Dieses ermöglicht die Manipulation mit Zahlen im Bereich von -128 bis +127 (oberstes Bit wird als Vorzeichen interpretiert).

Das Null-Flag (Z) wird gesetzt, wenn das infolge eines bestimmten Befehls entstandene Ergebnis der Wert Ø hat. Das Null-Flag wird gelöscht, wenn das Ergabnis nicht Ø ist. Ein Ergebnis, welches einen Übertrag hat, wobei jedoch im Akkumulator der Wert Ø steht, setzt sowohl das Carry-Flag, als auch das Null-Flag.

Hexadezimal	Binär
A7H	1 Ø 1 Ø Ø 1 1 1
+ 59H	+ Ø 1 Ø 1 1 Ø Ø 1
1ØØH	[1] Ø Ø Ø Ø Ø Ø Ø Ø

8 Bits vom Wert Ø setzen das Null-Flag auf 1

Carry-Bit = 1

Für das Inkrementieren oder Dekrementieren bestimmter Prozessor-Register zu dem Ergebnis Ø, wird ebenfalls das Null-Flag gesetzt.

Anschlußbezeichnungen und Funktionen

Bezeichnung (Symbol)	Nummer	Eingang (E) Ausgang (A)	Funktion
$A_8 - A_{15}$	21-28	A	Adressen-Bus: Die höherwertigen Speicheradressen oder die 8 Bits der E/A-Adresse; die Ausgänge sind während der Betriebszustände "Hold","Halt" und "Reset" hochohmig.
$AD_0 - AD_7$	12-19	E/A	Gemultiplexter Adressen-/Daten-Bus: Während der ersten Taktperiode (T-State) eines Maschinenzyklus stehen am Bus die niederwertigen 8 Bit der Speicheradresse (oder die E/A-Adresse) an. In der zweiten und dritten Taktperiode wird der Bus anschließend zum Datenbus.
ALE	30	A	Adressenspeicher-Freigabe: Das Signal tritt in der ersten Taktperiode eines Maschinezyklus auf und ermöglicht, daß die Adresse in den integrierten Adressenspeichern von Peripherie-Bausteinen gespeichert wird. Die fallende Flanke des ALE-Signals ist so eingestellt, daß die Aufbau- und Haltezeiten der Adresseninformation garantiert sind. Die fallende Flanke von ALE kann auch zum Takten der Status-Information verwendet werden. Der Ausgang ALE wird niemals hochohmig.

<table>
<tr><th>Bezeichnung (Symbol)</th><th>Nummer</th><th>Eingang (E) Ausgang (A)</th><th>Funktion</th></tr>
<tr><td>S_0, S_1, IO/$\overline{M}$</td><td>29, 33 34</td><td>A</td><td>Maschinenstatus
<table>
<tr><th>IO/M</th><th>S_1</th><th>S_0</th><th>Status</th></tr>
<tr><td>Ø</td><td>Ø</td><td>1</td><td>Speicher schreiben</td></tr>
<tr><td>Ø</td><td>1</td><td>Ø</td><td>Speicher lesen</td></tr>
<tr><td>1</td><td>Ø</td><td>1</td><td>E/A schreiben</td></tr>
<tr><td>1</td><td>1</td><td>Ø</td><td>E/A lesen</td></tr>
<tr><td>Ø</td><td>1</td><td>1</td><td>Op-Code-Abruf</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Interrupt-Quittung</td></tr>
<tr><td>*</td><td>Ø</td><td>Ø</td><td>Halt</td></tr>
<tr><td>*</td><td>X</td><td>X</td><td>Hold</td></tr>
<tr><td>*</td><td>X</td><td>X</td><td>Reset</td></tr>
</table>
* = hochohmig (Tri-state)
X = undefiniert

S_1 kann als voreilendes R/W-Statussignal verwendet werden. IO/$\overline{M}$, S_0 und S_1 werden am Anfang eines Maschinenzyklus gültig und bleiben es über den gesamten Zyklus. Die abfallende Flanke des ALE-Signals kann zum Speichern der Zustände dieser Leitung verwendet werden.</td></tr>
<tr><td>$\overline{RD}$</td><td>32</td><td>A</td><td>Lesen (Read): Ein L-Signal an $\overline{RD}$ gibt an, daß der ausgewählte Speicher- oder E/A-Baustein gelesen werden soll und daß der Daten-Bus für die Datenübertragung zur Verfügung steht. Der Ausgang ist in den Betriebsarten "Hold", "Halt" und "Reset" hochohmig.</td></tr>
</table>

Bezeichnung (Symbol)	Nummer	Eingang (E) Ausgang (A)	Funktion
$\overline{WR}$	31	A	Schreiben (Write): Ein L-Signal an $\overline{WR}$ gibt an, daß die am Daten-Bus anstehenden Daten in den angewählten Speicherplatz oder E/A-Kanal geschrieben werden sollen. Die Daten werden mit der steigenden Flanke von $\overline{WR}$ übergeben. Während der Betriebsarten "Hold", "Halt" und "Reset" ist der Ausgang hochohmig.
READY	35	E	Ein H-Signal an READY während eines Lese- oder Schreibzyklus zeigt an, daß der Speicher oder der Peripherie-Baustein zum Ausgeben oder Aufnehmen von Daten bereit ist. Wenn der READY-Eingang L-Signal führt, wartet die Zentraleinheit über eine ganze Zahl von Taktperioden, bis READY auf H-Pegel übergeht, worauf der Lese- oder Schreibzyklus beendet wird.
HOLD	39	E	HOLD zeigt an, daß eine andere Einheit den Adressen- oder Daten-Bus anfordert. Nach dem Eintreffen der HOLD-Anforderung gibt die Zentraleinheit den Bus frei, sobald die augenblicklich ablaufende Bus-Operation beendet ist. Interne Operationen können weiterhin ablaufen. Der Mikroprozessor hat erst dann wieder Zugang zum Bus, wenn das Signal HOLD zurückgenommen wird. Sobald das HOLD-Signal quittiert ist, werden die Adressen- und Daten-Leitungen sowie $\overline{RD}$, $\overline{WR}$ und IO/$\overline{M}$ hochohmig.

Bezeichnung (Symbol)	Nummer	Eingang (E) Ausgang (A)	Funktion
HLDA	38	A	Quittung von HOLD: Das Signal zeigt an, daß die Zentraleinheit eine HOLD-Anforderung empfangen hat und den Bus in der nächsten Taktperiode freigeben wird. HLDA geht auf L-Signal über, sobald die HOLD-Aforderung zurückgenommen ist. Die Zentraleinheit übernimmt den Bus eine halbe Taktperiode nach dem Übergang von HLDA auf L-Pegel.
INTR	10	E	Unterbrechungs-Anforderung: Der Eingang wird als allgemeiner Interrupteingang verwendet. Er wird nur während der vorletzten Taktperiode eines Befehls abgefragt sowie während der Zustände "Hold" und "Halt". Wenn der Eingang aktiv ist, wird das Inkrementieren des Programmzählers (PC) gesperrt und $\overline{\text{INTA}}$ ausgegeben. Während dieses Zyklus kann ein RESTART- oder CALL-Befehl für den Sprung zu dem Interrupt-Bedienungsprogramm eingefügt werden. Der Eingang INTR wird über Software freigegeben und gesperrt. Er wird ferner durch "Reset" sowie unmittelbar nach Annahme einer Unterbrechungsanforderung gesperrt.

Bezeichnung (Symbol)	Nummer	Eingang (E) Ausgang (A)	Funktion
$\overline{\text{INTA}}$	11	A	INTERRUPT-Quittung: Der Ausgang wird in den Befehlszyklus nach Annahme eines INTR-Signals anstelle von $\overline{\text{RD}}$ verwendet (und hat den gleichen Zeitablauf). Er kann zur Aktivierung des Unterbrechungs-Steuerbausteind SAB 8259A oder eines anderen Interrupt-Kanals verwendet werden.
RST 5.5 RST 6.5 RST 7.5	9 8 7	E	RESTART-Interrupt: Diese drei Eingänge haben den gleichen Zeitablauf wie INTR, bewirken jedoch, daß automatisch ein interner RESTART-Befehl eingefügt wird. Die Priorität dieser Interrupt-Eingänge ist so geordnet, wie in nachfolgender Tabelle gezeigt. Diese Interrupts haben eine höhere Priorität, als INTR. Außerdem können sie durch Verwendung des Befehls SIM einzeln maskiert werden.
TRAP	6	E	Der Interrupt-Eingang TRAP ist ein nicht maskierbarer RESTART-Interrupt-Eingang. Er wird gleichzeitig mit INTR oder RST 5.5-7.5 abgefragt und von keiner Maske oder Interrupt-Freigabe beeinflußt. Er hat von allen Interrupt-Eingängen die höchste Priorität (siehe nachfolgende Tabelle).

Bezeichnung (Symbol)	Nummer	Eingang (E) Ausgang (A)	Funktion
$\overline{\text{RESET IN}}$	36	E	Ein L-Signal an diesem Eingang setzt den Programmzähler auf Ø und das Interrupt-Freigabe- sowie HLDA-Flipflop zurück. Der Daten- und Adressen-Bus sowie die Steuerleitungen sind, solange RESET ansteht hochohmig. Da RESET asynchron arbeitet, können die internen Register und Flags des Prozessors durch RESET mit undefinierbaren Ergebnissen verändert werden. $\overline{\text{RESET IN}}$ ist ein Schmitt-Trigger-Eingang und ermöglicht den Anschluß einer R-C-Kombination zur RESET-Verzögerung beim Zuschalten der Versorgungsspannung. Die Zentraleinheit bleibt so lange zurückgesetzt, wie $\overline{\text{RESET IN}}$ ansteht.
RESET OUT	3	A	Zeigt an, daß die Zentraleinheit rückgesetzt wird. Dieser Ausgang kann als System-Rücksetzsignal verwendet werden. Das Signal ist mit dem Prozessor-Takt synchronisiert. Seine Dauer erstreckt sich über eine ganze Zahl von Taktperioden.
X_1, X_2	1, 2	E	An X_1 und X_2 wird ein Quarz, eine LC- oder eine RC-Kombination zur Anregung des internen Taktgenerators angeschlossen. X_1 kann auch als Eingang für einen externen Takt verwendet werden. Die Eingangsfrequenz wird halbiert und ergibt die interne Arbeitsfrequenz des Prozessors.

Bezeichnung (Symbol)	Nummer	Eingabe (E) Ausgabe (A)	Funktion
CLK	37	A	Taktausgang zur Verwendung als Systemtakt. Die Periodendauer des Signals CLK beträgt das Doppelte der Eingangs-Periodendauer an X_1, X_2.
SID	5	E	Eingang für serielle Datenübertragung. Der Zustand dieses Eingangs wird bei jeder Ausführung eines RIM-Befehls in das Bit 7 des Akkumulators übertragen.
SOD	4	A	Ausgang für serielle Datenübertragung. Der Ausgang SOD wird entsprechend der Form des Befehls SIM gesetzt bzw. rückgesetzt.
V_{cc}	40	-	Versorgungsspannung (+5 V)
V_{ss}	20	-	Masse (0 V)

Unterbrechungs-Prioritäten, Restart-Adressen und Ansprechbedingungen

Bezeichnung	Priorität	Sprungadresse bei Auftreten der Unterbrechungsanforderung 1)	Art der Triggerung
TRAP	1	24H	Ansteigende Flanke und H-Pegel bis zur Abfrage
RST 7.5	2	3CH	Ansteigende Flanke (wird gespeichert)
RST 6.5	3	34H	H-Pegel bis zur Abfrage
RST 5.5	4	2CH	H-Pegel bis zur Abfrage
INTR	5	2)	H-Pegel bis zur Abfrage

1) Vor dem Sprung zu der angegebenen Adresse, legt der Prozessor den Programmzählerstand (PC) im Stack (Stapelspeicher) ab.

2) Die Sprungadresse hängt von dem Befehl ab, welcher der Zentraleinheit bei Quittung der Unterbrechnugsanforderung gegeben wird.

3.2 Programmierbarer Interface-Baustein SAB 8255 A

Über den programmierbaren Interface-Baustein SAB 8255 A ist der Datenverkehr zwischen Mikrokomputer und peripheren Komponenten möglich. Dazu besitzt der Baustein drei 8-Bit-Kanäle (Ports), die vom Programm wahlweise auf Ein- oder Ausgabe-Betrieb geschaltet werden können. Die zu übertragenden Informationen kommen vom Daten-Bus bzw. gelangen dort hin. Über einen Steuerbefehl wird dem Interface-Baustein die gewünschte Betriebsart (MODE Ø, 1, 2) mitgeteilt, die bis zum Eintreffen einer neuen Anweisung beibehalten wird. Die Betriebsarten bedeuten:

MODE Ø: Einfache Ein-/Ausgabe
MODE 1: Getastete Ein-/Ausgabe
MODE 2: Bidirektionale Ein-/Ausgabe.

Um die Funktion der einzelnen Kanäle festzulegen, muß ein Datenwort (Steuerwort) eingeschrieben werden. Der Kanal C ist in zwei 4-Bit-Kanäle geteilt. Dadurch ergeben sich 16 Variationen für die Ein-/Ausgabefestlegung.

MODE Ø, einfache Ein-/Ausgabe

Diese Funktions-Anordnung ermöglicht eine einfache Eingabe und Ausgabe für jeden der drei Kanäle. Es ist kein Qiuttungsaustausch erforderlich, denn Daten werden einfach in den ausgewählten Kanal geschrieben oder aus ihm gelesen.

Prinzipielle Funktionsdefinition von MODE Ø:

- Zwei 8-Bit-Kanäle und zwei 4-Bit-Kanäle
- Jeder Kanal kann Eingang oder Ausgang sein
- Ausgänge haben Zwischenspeicher
- Eingänge arbeiten ohne Zwischenspeicher
- 16 verschiedene Ein-/Ausgabe-Kombinationen sind möglich.

Definition der Steuerworte für MODE Ø

Gruppe A Kanal A PA_7-$PA_Ø$	Gruppe A Kanal C PC_7-PC_4	Gruppe B Kanal B PB_7-$PB_Ø$	Gruppe B Kanal C PC_3-$PC_Ø$	Steuerwort	Hex-Code
A	A	A	A	1 Ø Ø Ø Ø Ø Ø Ø	8Ø
A	A	A	E	1 Ø Ø Ø Ø Ø Ø 1	81
A	A	E	A	1 Ø Ø Ø Ø Ø 1 Ø	82
A	A	E	E	1 Ø Ø Ø Ø Ø 1 1	83
A	E	A	A	1 Ø Ø Ø 1 Ø Ø Ø	88
A	E	A	E	1 Ø Ø Ø 1 Ø Ø 1	89
A	E	E	A	1 Ø Ø Ø 1 Ø 1 Ø	8A
A	E	E	E	1 Ø Ø Ø 1 Ø 1 1	8B
E	A	A	A	1 Ø Ø 1 Ø Ø Ø Ø	9Ø
E	A	A	E	1 Ø Ø 1 Ø Ø Ø 1	91
E	A	E	A	1 Ø Ø 1 Ø Ø 1 Ø	92
E	A	E	E	1 Ø Ø 1 Ø Ø 1 1	93
E	E	A	A	1 Ø Ø 1 1 Ø Ø Ø	98
E	E	A	E	1 Ø Ø 1 1 Ø Ø 1	99
E	E	E	A	1 Ø Ø 1 1 Ø 1 Ø	9A
E	E	E	E	1 Ø Ø 1 1 Ø 1 1	9B

MODE 1, getastete Ein-/Ausgabe

Diese Funktionsanordnung dient zum Austausch von Ein-/Ausgabedaten zu oder von einem ausgewählten Kanal in Verbindung mit Abtastimpulsen oder "Quittungs"-Signalen. Kanal A und Kanal B benützt in der Betriebsart 1 die Leitungen des Kanals C, um diese"Quittungs"-Signale zu erzeugen oder zu empfangen.

Prinzipielle Funktionsdefinition von MODE 1

- Zwei Gruppen von Kanälen
- Jede Gruppe umfaßt einen 8-Bit-Datenkanal und einen 4-Bit-Steuer-/Datenkanal
- Der 8-Bit-Datenkanal kann entweder als Eingang oder Ausgang verwendet werden. Die Ein- und Ausgangsdaten werden zwischengespeichert
- Der 4-Bit-Kanal wird für Steuer- und Zustandszwecke für die 8-Bit-Datenkanäle benutzt. Anschlüsse, die nicht für Steuer- und Zustandszwecke benötigt werden, sind für Ein-/Ausgabe-Operationen frei. Eingabedaten werden in Kanal C nicht zwischengespeichert.

MODE 2, getastete Zweiweg-Bus-Ein-/Ausgabe)

Diese Funktionsanordnung ermöglicht den Datenaustausch mit einem Peripheriegerät oder einer Schaltung auf einem einzigen 8-Bit-Bus, über den Daten gesendet und empfangen werden. Der richtige Datenfluß auf dem Bus wird in ähnlicher Weise wie bei MODE 1 durch Quittungs-Signale gewährleistet. Die Erzeugung von Unterbrechungen und die Funktion Sperren/Freigeben stehen ebenfalls zur Verfügung.

Prinzipielle Funktionsdefinition von MODE 2

- Nur in Gruppe A verwendet
- Ein 8-Bit-Zweiweg-Bus-Kanal (Kanal A) und ein 5-Bit-Steuerkanal (Kanal C)
- Eingänge und Ausgänge verfügen über Pufferspeicher
- Der 5-Bit-Steuerkanal (Kanal C) wird für Steuer- und Zustandszwecke für den 8-Bit-Zweiweg-Bus-Kanal verwendet.

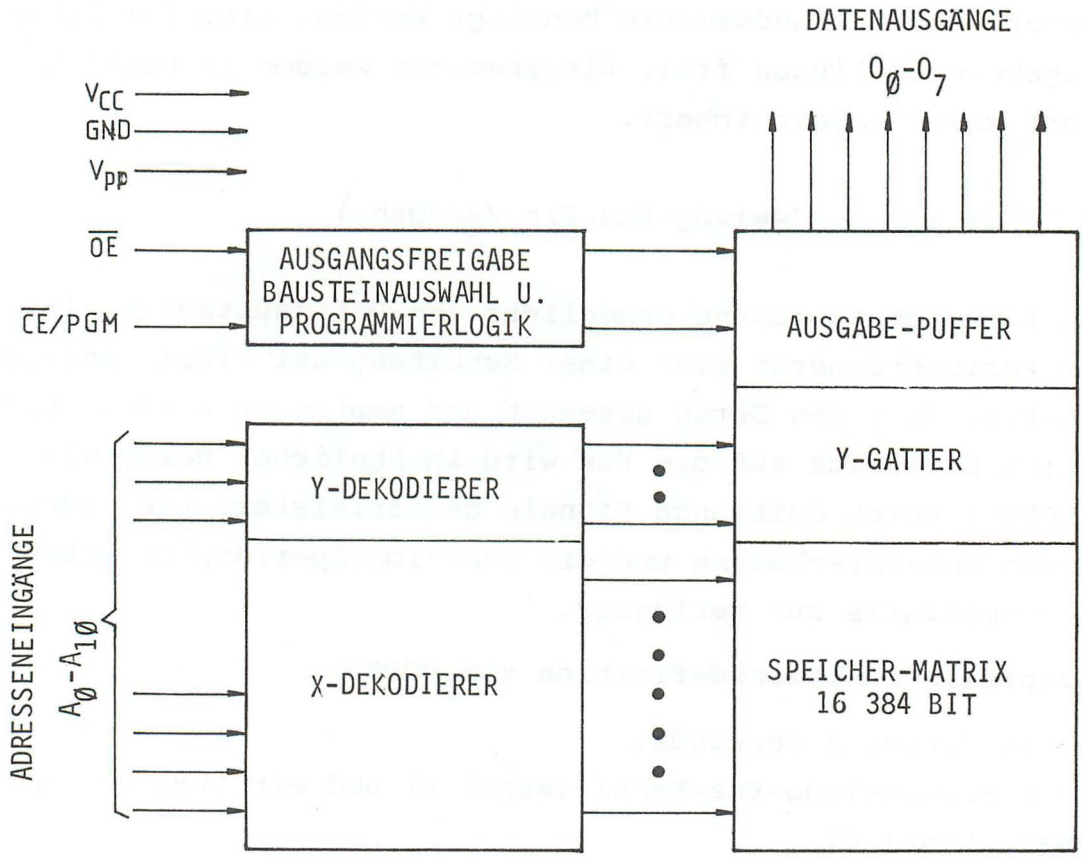
3.3 Festwertspeicher EPROM 16 K SAB 2716

Der SAB 2716 ist ein mit UV-Strahlung löschbarer und elektrisch programmierbarer Festwertspeicher für 16 384 Bit, organisiert zu 2 048 Byte (2KByte). Er besitzt eine statische Warte-Betriebsart (Standby) und ermöglicht ein schnelles Programmieren von einzelnen Adressen.

Statisches Standby setzt die Verlustleistung herab, ohne die Zugriffszeit zu erhöhen. Die Verlustleistung beträgt im aktiven Zustand 525 mW (max.), im Standby-Zustand 132 mW (max.), die Zugriffszeit 450 ns (max.).

Der SAB 2716 wird mit einem einzelnen Impuls im TTL-Pegel programmiert. Mit der Einzeladressen-Programmierung des SAB 2716 kann jeder Speicherplatz jederzeit programmiert werden, entweder einzeln, sequenziell oder wahlfrei.

Blocksch altbild



Wahl der Betriebsart

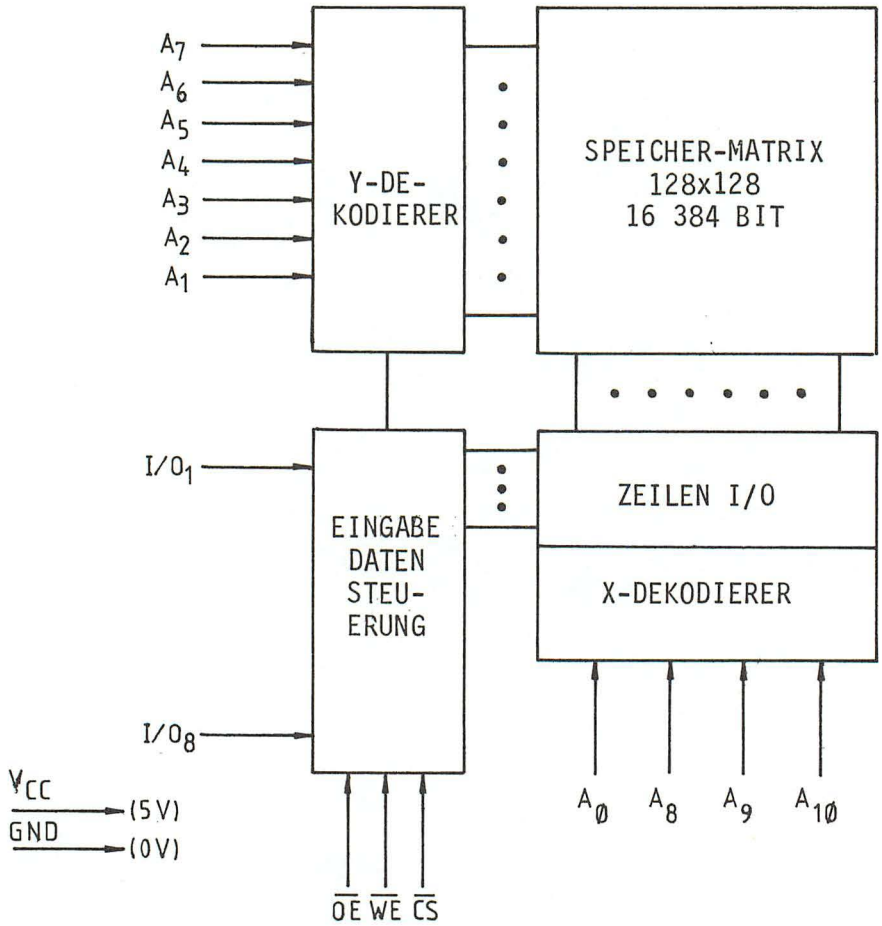
Anschlüsse / Betriebsart	$\overline{CE}$/PGM	$\overline{OE}$	V_{pp}	V_{CC}	Ausgänge
Lesen	V_{IL}	V_{IL}	+5 V	+5 V	D_{OUT}
Standby (Wartebetrieb)	V_{IH}	bedeutungslos	+5 V	+5 V	hochohmig
Programmieren	V_{IL} auf V_{IH} gepulst	V_{IH}	+25 V	+5 V	D_{IN}
Kontrolle der Programmierung	V_{IL}	V_{IL}	+25 V	+5 V	D_{OUT}
Sperre der Programmierung	V_{IL}	V_{IH}	+25 V	+5 V	hochohmig

3.4 Schreib-/Lese-Speicher-Baustein CMOS-RAM HM 6116 LP

RAMs sind Speicher, deren einzelne Speicherzellen als bistabile Flipflop-Schaltungen ausgelegt sind, wodurch die gespeicherte Information solange erhalten bleibt, wie die Versorgungsspannung anliegt. Durch die CMOS-Technologie nehmen sie im Ruhezustand einen vernachlässigbaren Strom auf. Dadurch können sie leicht durch Batterien gepuffert werden und eignen sich hervorragend zur Datensicherung bei Versorgungsspannungs-Ausfall.

Der CMOS-RAM HM 6116 LP ist für einen Speicherbereich von 16 384 Bit zu 8 Bit, also 2 048 Byte (2KByte). Er ist durch Brücke kompatibel mit dem Festwertspeicher SAB 2716 und direkt TTL-kompatibel. Die Verlustleistung beträgt im aktiven Zustand 160 mW (typ), im Standby-Zustand 10 µW (typ), die Zugriffszeit 200 ns (max.).

Blockschaltbild



Wahl der Betriebsart

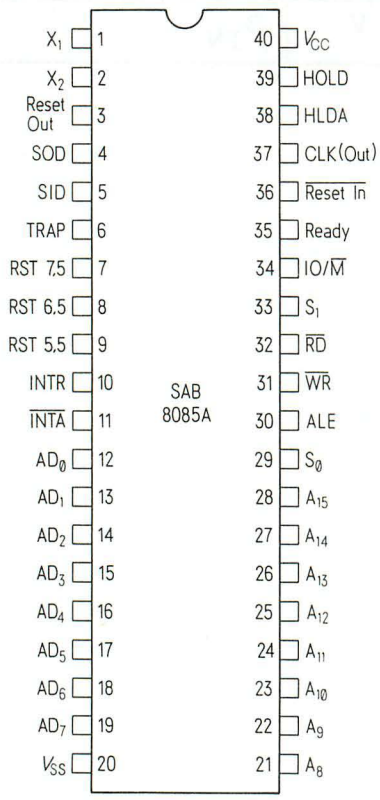
Anschlüsse / Betriebsart	$\overline{CS}$ (18)	$\overline{OE}$ (20)	$\overline{WE}$ (21)	V_{CC} (24)	I/O (9-11,13-17)
Standby	H	X	X	+5 V	hochohmig
Lesen	L	L	H	+5 V	D_{OUT}
Schreiben	L	H	L	+5 V	D_{IN}
Schreiben	L	L	L	+5 V	D_{IN}

4 Gesamtübersicht der Bauteile und der Anschlußbelegung

4.1 CPU-Grundplatine

4.1.1 SAB 8085 A 8-Bit Mikroprozessor

Anschlußbelegung



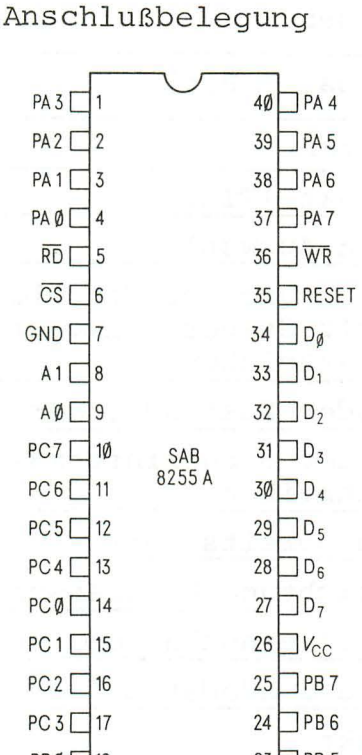
Anschlußbezeichnungen

A_8-A_{15}	Adressen-Bus (8 Bit)
$AD_Ø$-AD_7	Gemultiplexter Adressen-/ Daten-Bus (8 Bit)
ALE	Adressenspeicher-Freigabe
$S_Ø$, S_1, IO/$\overline{M}$	Maschinenstatus
$\overline{RD}$	Daten oder Zustand lesen
$\overline{WR}$	Daten oder Steuerinformation Schreiben
READY	Prozessor bereit
HOLD	Adressen- und Daten-Bus-anforderung
HLDA	HOLD-Quittierung
INTR*	Unterbrechungsanforderung
$\overline{INTA}$	Unterbrechungs-Quittierung
RST 5.5* ... RST 7.5*	RESTART-Unterbrechung
$\overline{RESET\ IN}$	Rücksetzen Zentraleinheit
RESET OUT	Rücksetzen System
X_1, X_2	Eingang Systemtakt
CLK	Ausgang Systemtakt
SID	Eingang serielle Datenübertragung
SOD	Ausgang serielle Datenübertragung
V_{CC}	Versorgungsspannung (5 V)
V_{ss}	Masse (0 V)
TRAP*	Unterbrechungs-Eingang

* Unterbrechungs-Prioritäten in Kapitel 3.1

4.1.2 SAB 8255 A, programmierbarer peripherer Schnittstellen-Baustein

Anschlußbelegung

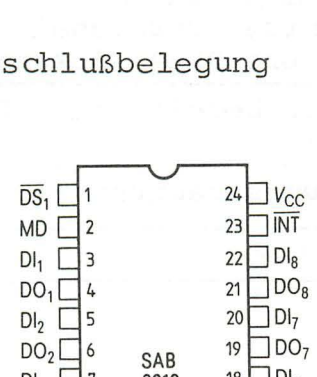


Anschlußbezeichnungen

$D_Ø - D_7$	Daten-Bus (Zweiweg)
RESET	Rücksetz-Eingang
$\overline{CS}$	Baustein-Auswahl
$\overline{RD}$	Lese-Eingang
$\overline{WR}$	Schreib-Eingang
AØ, A1	Kanal-Adressen
$PA_Ø - PA_7$	Kanal A (Bit Ø bis Bit 7)
$PB_Ø - PB_7$	Kanal B (Bit Ø bis Bit 7)
$PC_Ø - PC_3$	Kanal C niederwertig
$PC_4 - PC_7$	Kanal C höherwertig
V_{CC}	Versorgungsspannung (5 V)
GDN	Masse (O V)

4.1.3 SAB 8212, 8-Bit-Ein-/Ausgabe-Baustein

Anschlußbelegung

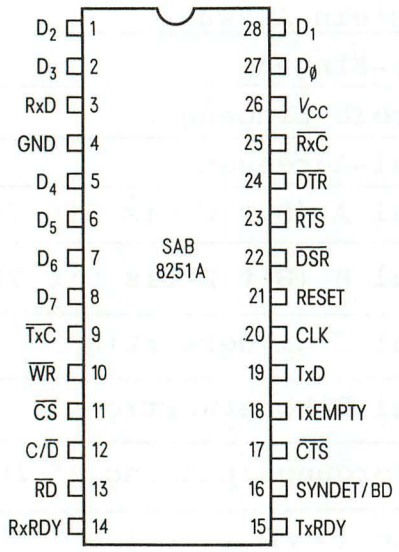


Anschlußbezeichnungen

$DI_1 - DI_8$	Daten-Eingänge
$DO_1 - DO_8$	Daten-Ausgänge
$\overline{DS_1}$, DS_2	Baustein-Auswahl
MD	Betriebsart
STB	Übernahme
$\overline{INT}$	Unterbrechung
$\overline{CLR}$	Löschen
V_{CC}	Versorgungsspannung (5 V)
GND	Masse (O V)

4.1.4 SAB 8251 A, programmierbarer Serien-Schnittstellen-Baustein

Anschlußbelegung



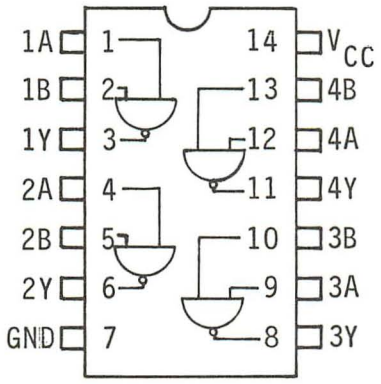
Anschlußbezeichnungen

$D_{Ø}$-D_7	Daten-Bus (8 Bit)
RESET	Rücksetzen
CLK	Systemtakt (TTL)
$\overline{CS}$	Baustein-Auswahl
C/$\overline{D}$	Kennzeichnung der Parallel-Daten als Steuer- bzw. Status-Wort/Zeichen
$\overline{RD}$	Daten oder Zustand lesen
$\overline{WR}$	Daten oder Steuerinformation schreiben
$\overline{DSR}$	Betriebsbereitschaft
$\overline{DTR}$	DE-Einrichtung betriebsbereit
$\overline{RTS}$	Sendeteil einschalten
TxD	Serielle Sendedaten
$\overline{TxC}$	Sendetakt
$\overline{CTS}$	Sender-Freigabe
TxEMPTY	Senderegister leer
TxRDY	Sender bereit (kann Daten annehmen)
RxD	Serielle Empfangsdaten
$\overline{RxC}$	Empfangstakt
SYNDET/BD	Synchronisationssignal/-erkennung und BREAK-Erkennung
RxRDY	Empfänger bereit (kann Daten liefern)
V_{CC}	Versorgungsspannung (+5 V)
GND	Masse (0 V)

4.1.5 SN74LSOON, Vier-NAND-Gatter mit je 2 Eingängen

P_D = 8 mW (typ), t_{pd} = 9,5 ns (typ)

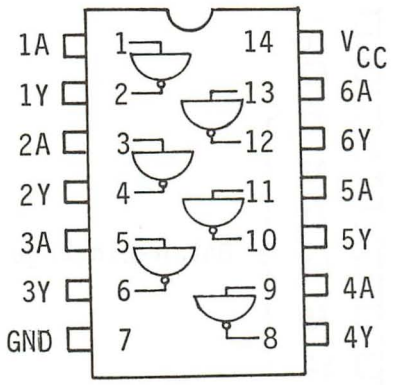
Anschlußbelegung



4.1.6 SN74LSO4N, Sechs Inverter

P_D = 12 mW (typ), t_{pd} = 9,5 ns (typ)

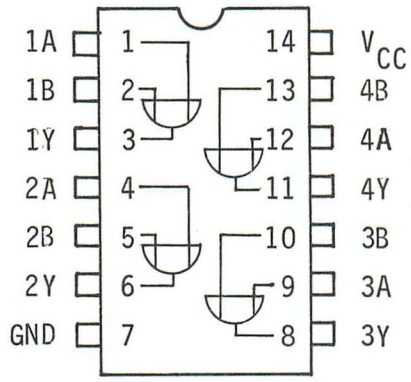
Anschlußbelegung



4.1.7 SN74LS32N, vier OR-Gatter mit 2 Eingängen

P_D = 20 mW (typ), t_{pd} = 14 ns (typ)

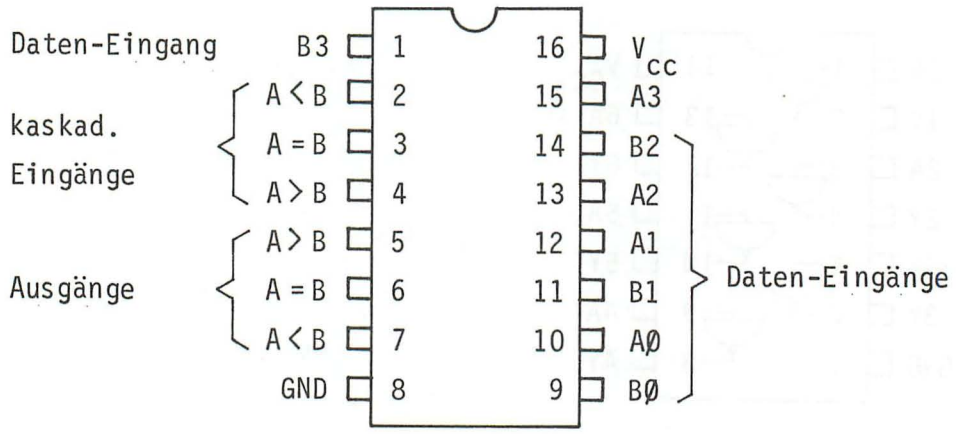
Anschlußbelegung



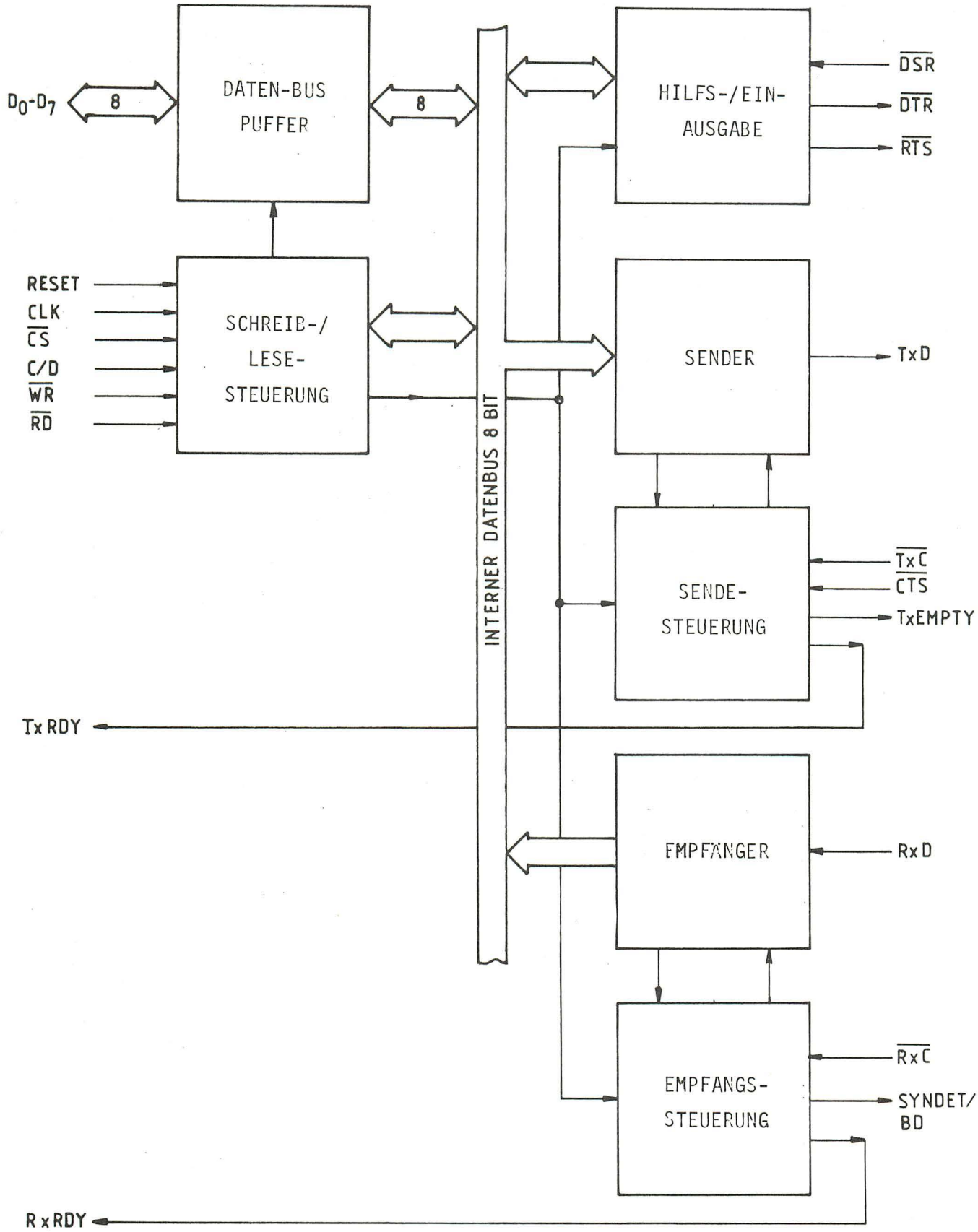
4.1.8 SN74LS85N, 4-Bit-Vergleicher

P_D = 52 mW (typ), t_{pd} = 23,5 ns (typ)

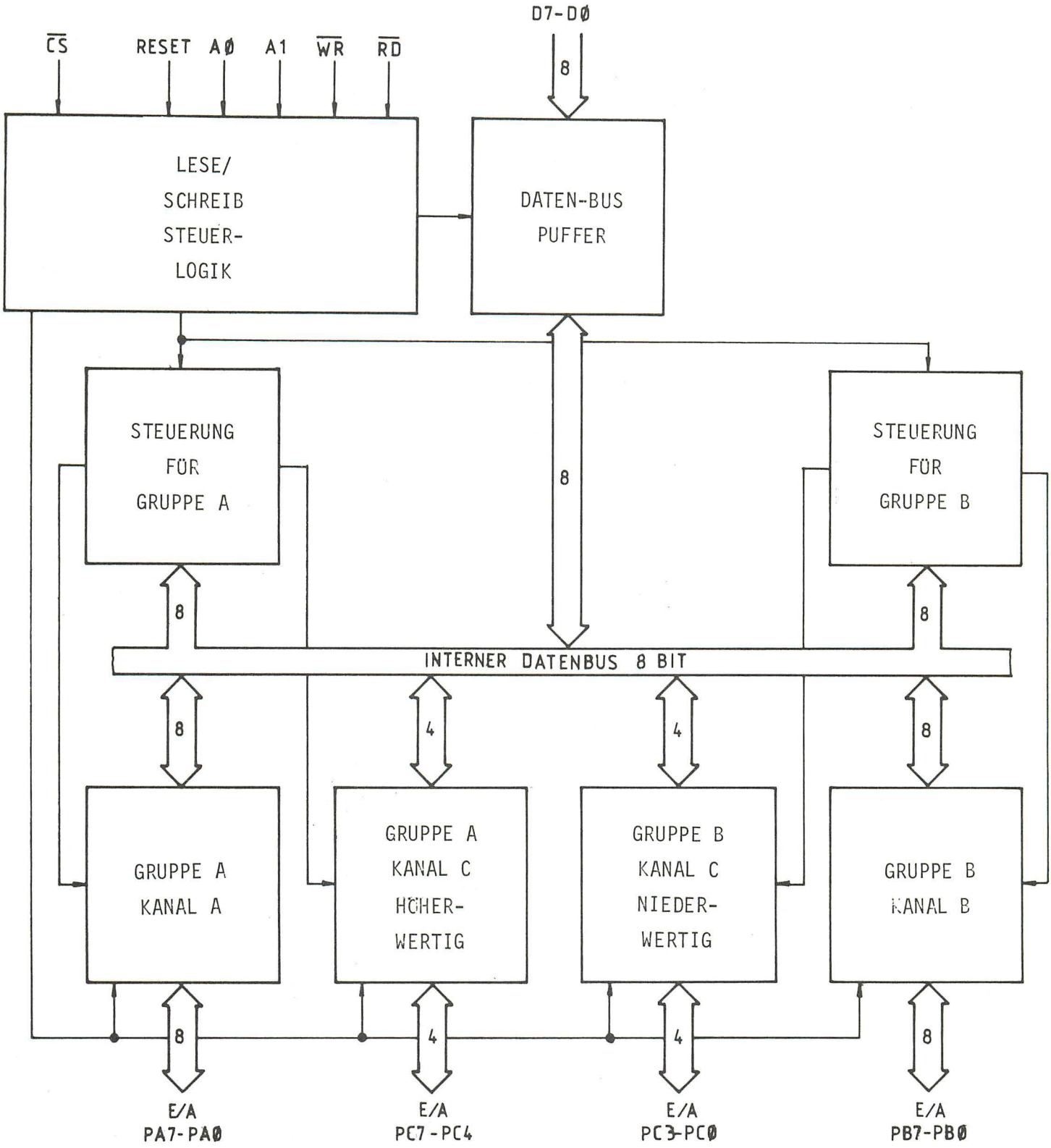
Anschlußbelegung



Blockschaltbild



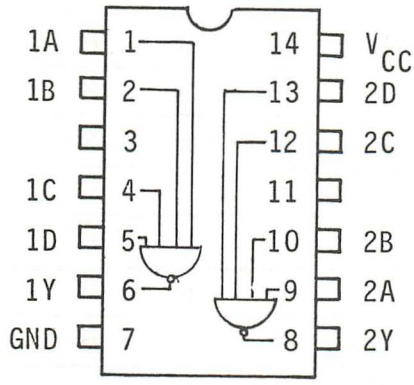
Blockschaltbild



4.1.9 SN74LS2ON, Zwei NAND-Gatter mit je 4 Eingängen

P_D = 4 mW (typ), t_{pd} = 9,5 ns (typ)

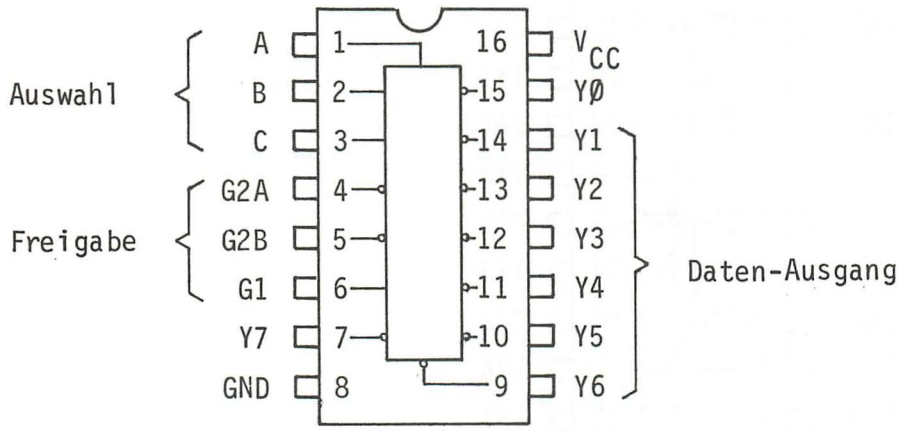
Anschlußbelegung



4.1.10 SN74L138N, 3-Bit-Binärdekoder/Demultiplexer (3 zu 8)

P_D = 32 mW (typ), t_{pd} = 22 ns (typ)

Anschlußbelegung

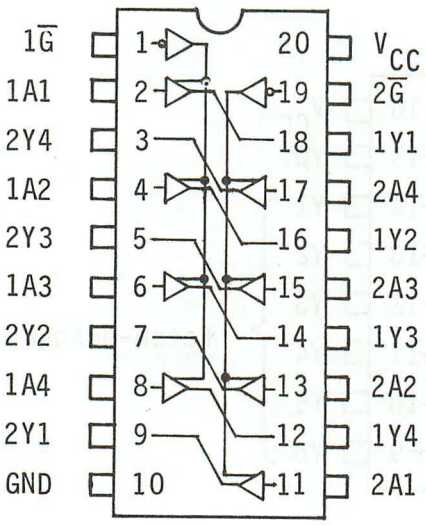


Wahrheitstabelle

Eingänge					Ausgänge							
Freigabe		Auswahl										
G1	G2	C	B	A	YØ	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

4.1.11 SN74LS244, Treiberstufe mit Tri-State-Ausgängen

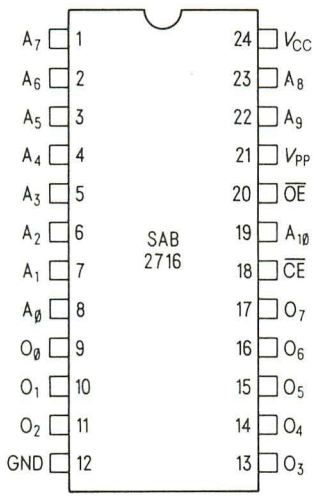
Anschlußbelegung



4.2 Speicherplatine 16 K ROM (RAM)

4.2.1 SAB 2716, EPROM-Festwertspeicher

Anschlußbelegung

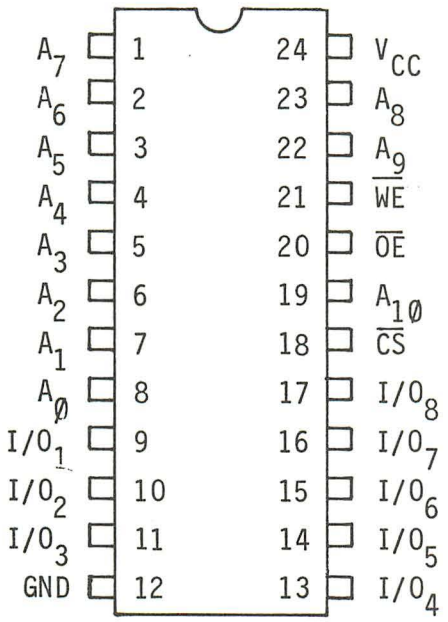


Anschlußbezeichnungen

$A_{Ø}$-$A_{1Ø}$	Adressen
$\overline{CE}$/PGM	Baustein-Freigabe/Programmierung
$\overline{OE}$	Ausgangs-Freigabe
$O_{Ø}$-O_7	Ausgänge
V_{CC}	Versorgungsspannung (+5 V)
V_{PP}	Versorgungsspannung (+5 V) Programmierspannung (+25 V)
GND	Masse (O V)

4.2.2 HM 6116 LP, CMOS-Schreib-/Lesespeicher

Anschlußbelegung



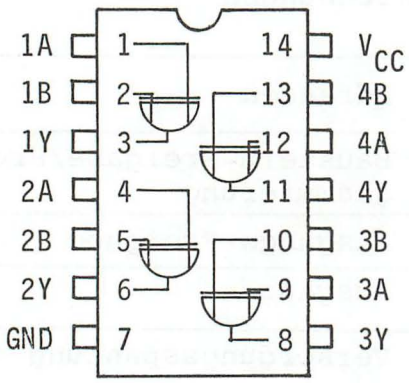
Anschlußbezeichnungen

$A_{Ø}$-$A_{1Ø}$	Adressen
$\overline{CS}$	Baustein-Freigabe/Programmierung
I/O_1-I/O_8	Ausgänge
V_{CC}	Versorgungsspannung (+5 V)
$\overline{WE}$	Versorgungsspannung (+5 V) Versorgungsspannung (+25 V)
GND	Masse (O V)
$\overline{OE}$	Ausgangs-Freigabe

4.2.3 SN74LS86N, Vier EXCLUSIVE-OR-Gatter mit je 2 Eingängen

P_D = 30 mW (typ), t_{pd} = 10 ns (typ)

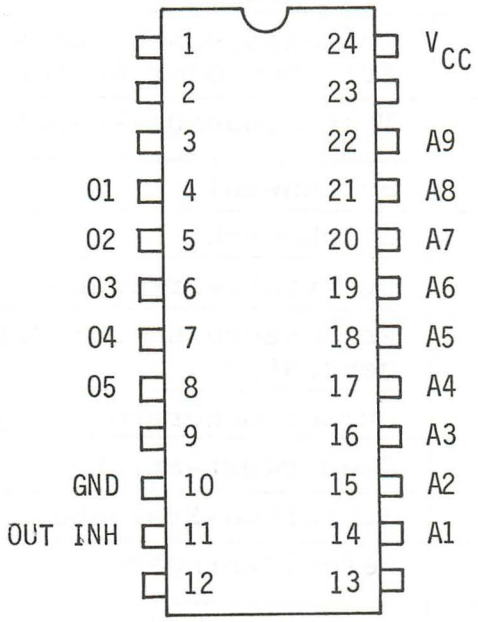
Anschlußbelegung



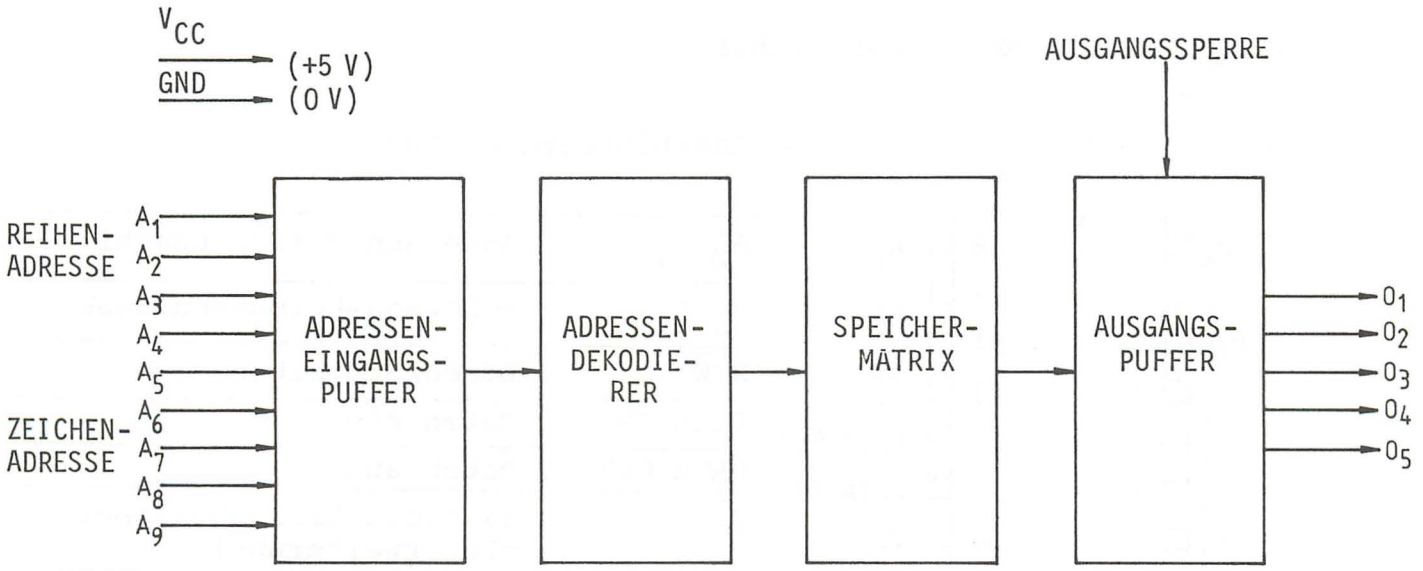
4.3 RAM/Video-Grundplatine

4.3.1 RO-3-2513, Zeichengenerator (stat. ROM), vorprogrammiert für Standard-ASCII-Code, organisiert als 512-Fünf-Bit-Worte, 5x7-Punkt-Matrix

Anschlußbelegung

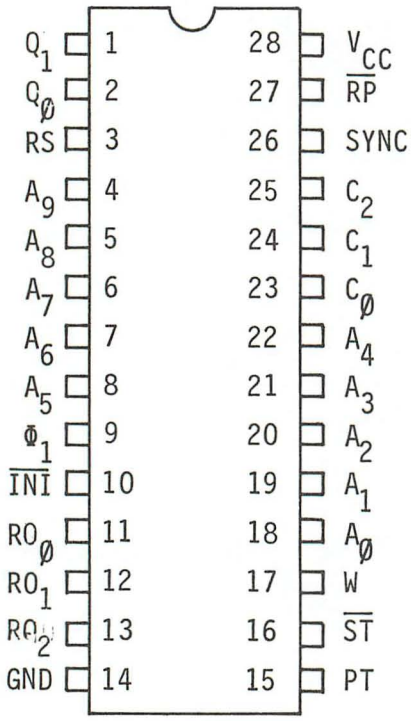


Blockschaltbild



4.3.2 EF9364A, Bildschirm-Steuersignal-Baustein zur Steuerung von 16 Zeilen zu 64 Zeichen einer Bildschirm-Anzeige; mit Zeilenauffrischung, Zeicheneingang und Cursorsteuerung.

Anschlußbelegung

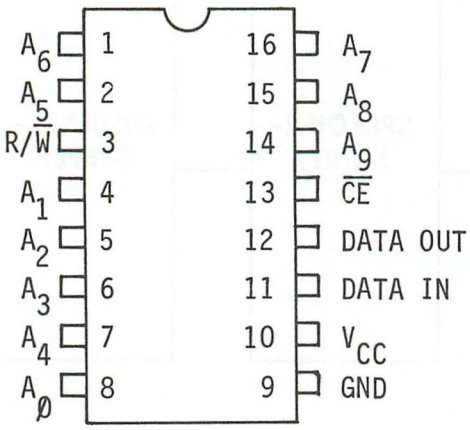


Anschlußbezeichnungen

$A_\emptyset$-A_4	Adressen der Zeichenpositionen
A_5-A_9	Zeilenadressen zum Auffrischen oder Schreiben
$Q_\emptyset$, Q_1	Takt-Eingang/-Ausgang
RS	Seitenwahl
Φ	Zeichentakt
$\overline{INI}$	Punkttakt-Freigabe
$RO_\emptyset$-RO_2	Reihenadresse für Zeichengenerator
PT	Cursor-Generator-Ausgang
$\overline{ST}$	Steuerwort-Strobe
W	Schreiben-Freigabe
$C_\emptyset$-C_2	Befehls-Eingänge
SYNC	Video-Synchronisierung
$\overline{RP}$	Ende der Seite
V_{CC}	Versorgungsspannung (+5 V)
GND	Masse (0 V)

4.3.3 MM2102, MOS-RAM-Speicher

Anschlußbelegung

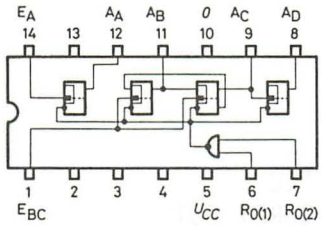


Anschlußbezeichnungen

$A_\emptyset$-A_4	Adressen-Zeilenauswahl
A_5-A_9	Adressen-Kolonnenauswahl
$R/\overline{W}$	Lesen/Schreiben
DATA IN	Daten ein
DATA OUT	Daten aus
$\overline{CE}$	Baustein-Freigabe (Speichererweiterung)
V_{CC}	Versorgungsspannung (+5 V)
GND	Masse (0 V)

4.3.4 DM7492

Anschlußbild, Draufsicht



Funktionstabellen

Binäruntersetzung 1:12
(A_A mit E_{BC} verbunden)

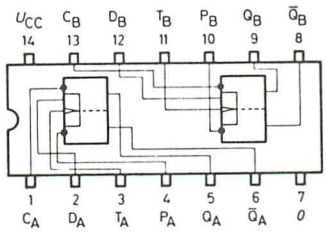
Zählschritt	A_D	A_C	A_B	A_A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	H	L	L	L
7	H	L	L	H
8	H	L	H	L
9	H	L	H	H
10	H	H	L	L
11	H	H	L	H

Rückstellen des Zählers

$R_{0(1)}$	$R_{0(2)}$	A_D	A_C	A_B	A_A
H	H	L	L	L	L
L	L	zählt			
L	H	zählt			
H	L	zählt			

4.3.5 DM7474, Dual-D-Flipflop

Anschlußbild, Draufsicht



Funktionstabellen

Eingang D

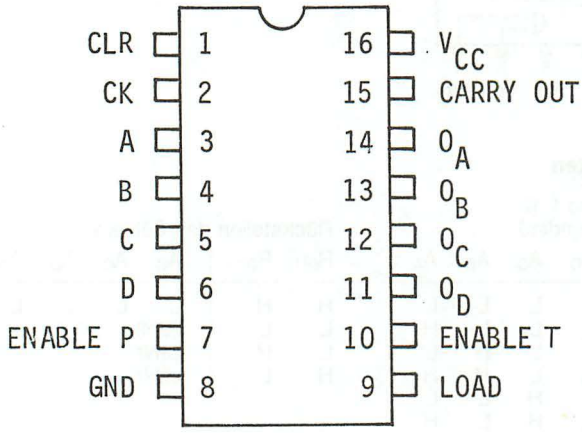
$t = t_n$	$t = t_{n+1}$	
D	Q	$\overline{Q}$
L	L	H
H	H	L

direkte Eingänge

P	C	Q	$\overline{Q}$
L	H	H	L
H	L	L	H
H	H	?	?
L	L	H	H

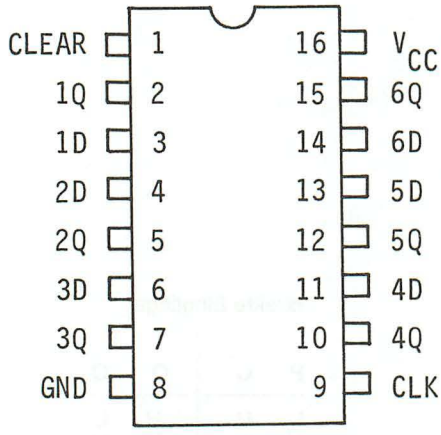
4.3.6 DM74163, 4-Bit-Binärzähler

Anschlußbelegung



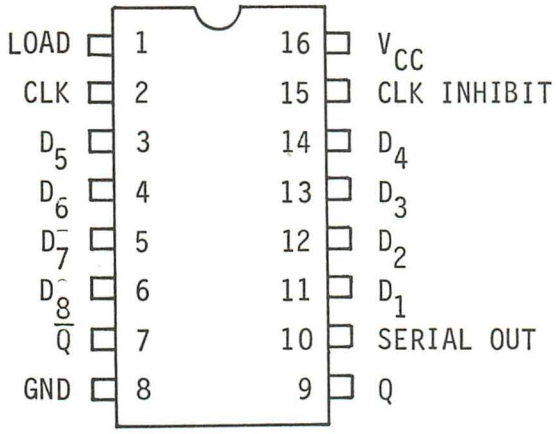
4.3.7 SN74LS174N, Sechs D-Flipflops mit gemeinsamem Clear

Anschlußbelegung

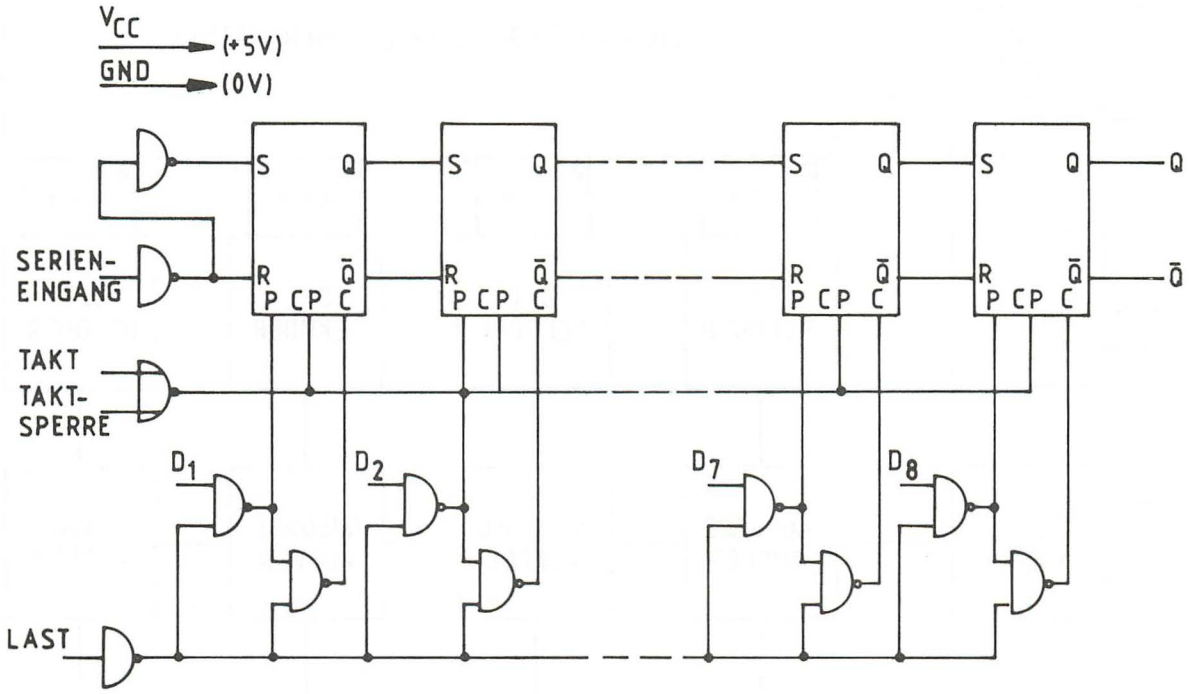


4.3.8 SN74165, 8-Bit-Schieberegister mit parallelem Eingang und seriellem Ausgang

Anschlußbelegung

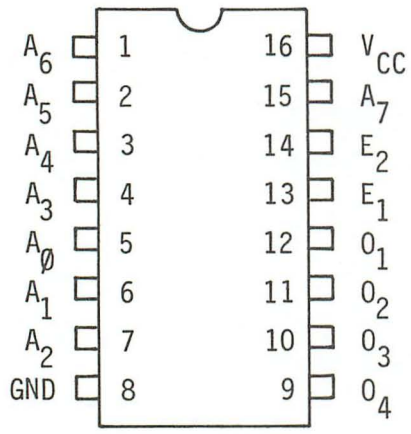


Blockschaltbild

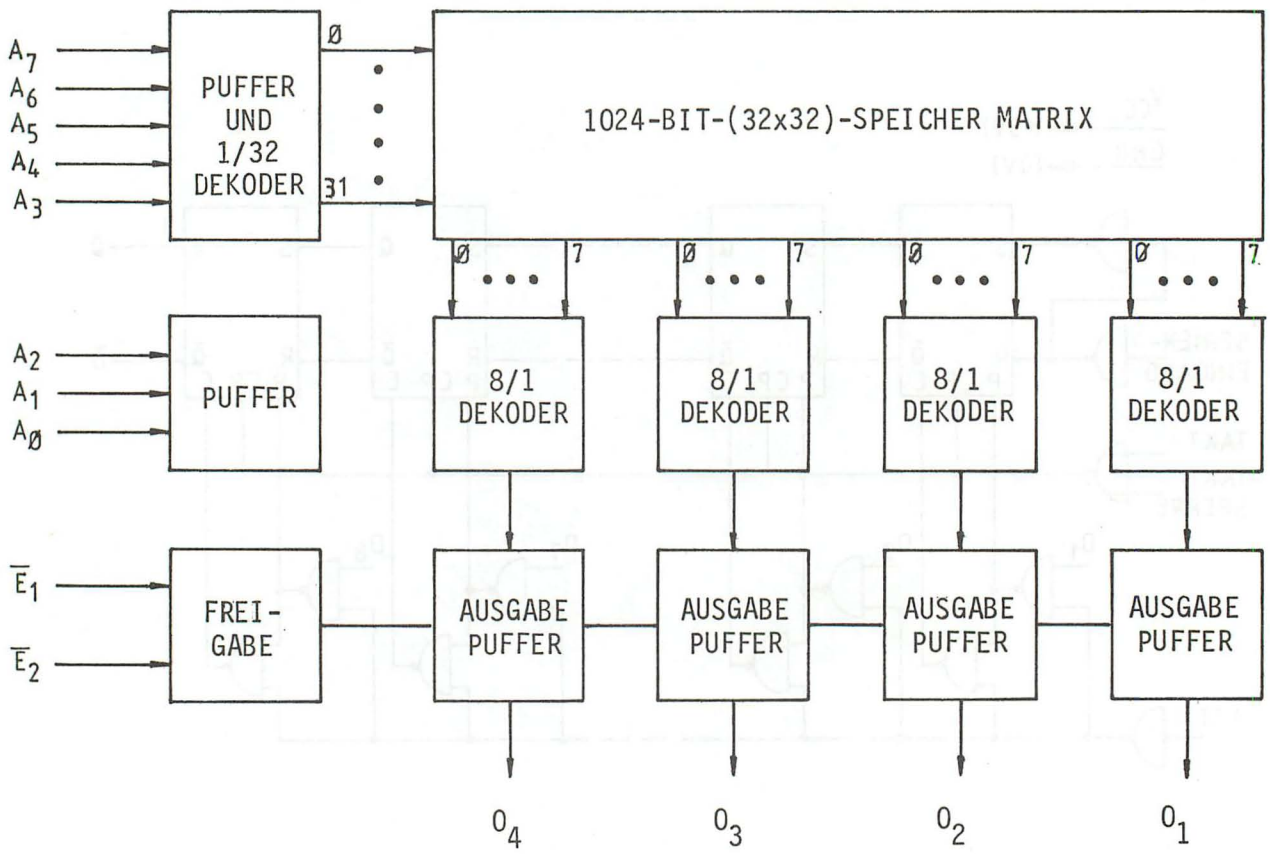


4.3.9 DM74S287/387, Bipolar PROM

Anschlußbelegung



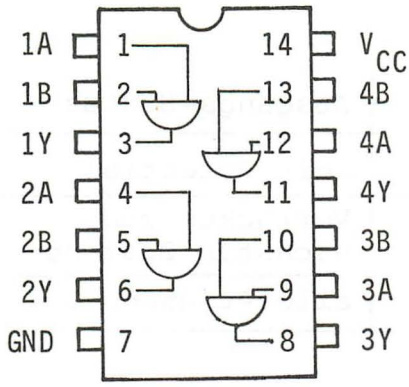
Blockschaltbild



4.3.10 SN74LS08N, Vier UND-Gatter mit je 2 Eingängen

P_D = 17 mW (typ), t_{pd} = 12 ns (typ)

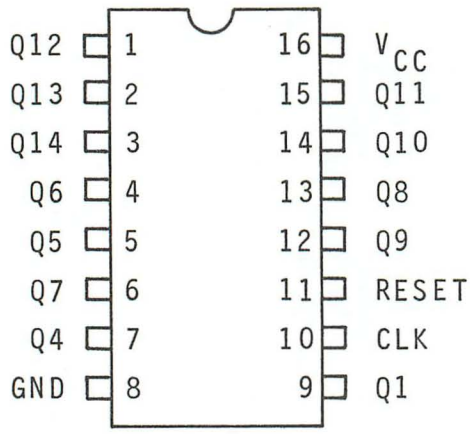
Anschlußbelegung



4.3.11 Kassetteninterface

4.3.11.1 MC14020B, 14-Bit-Binärzähler

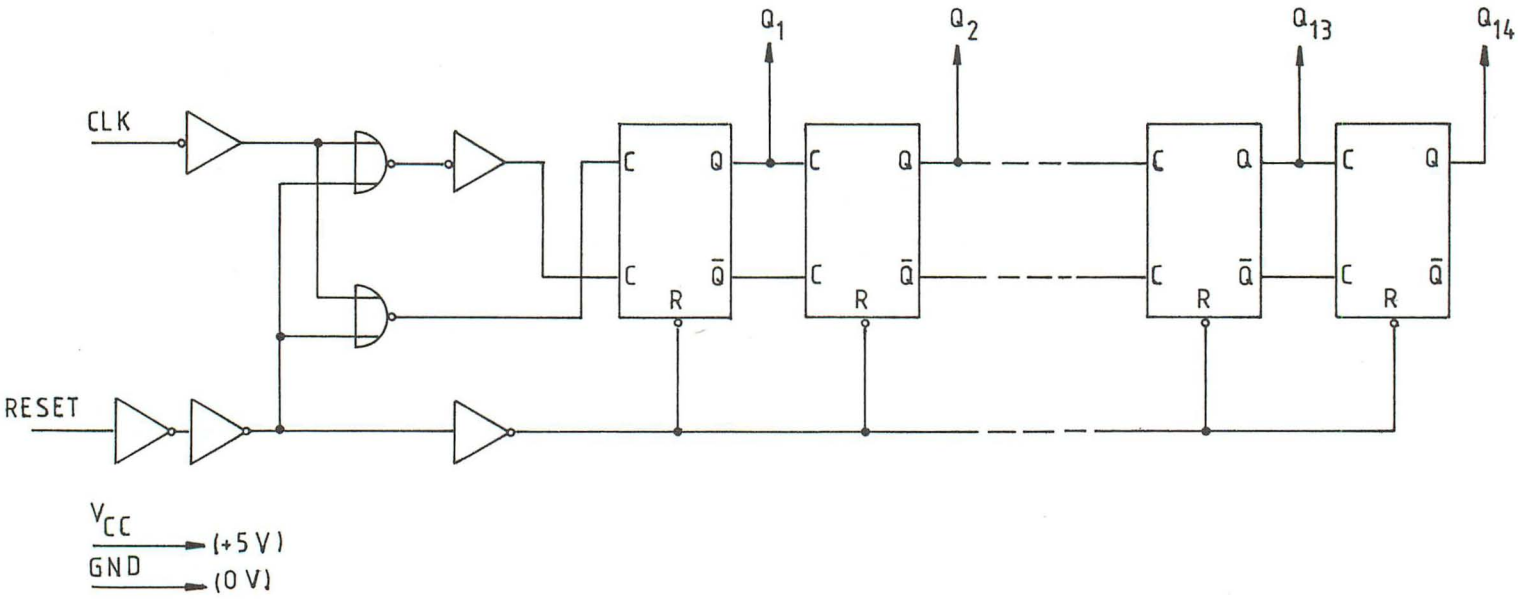
Anschlußbelegung



Wahrheitstabelle

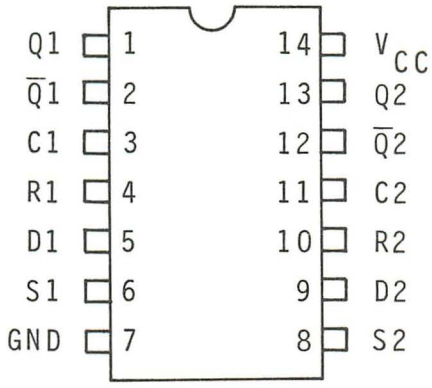
CLK	RESET	Ausgangs-Status
↗	Ø	kein Potential
↘	Ø	Vorrücken zum nächsten Zustand
X	1	alle Ausgänge L

Blockschaltbild



4.3.11.2 MC14013B, Zweifach-D-Flipflop

Anschlußbelegung

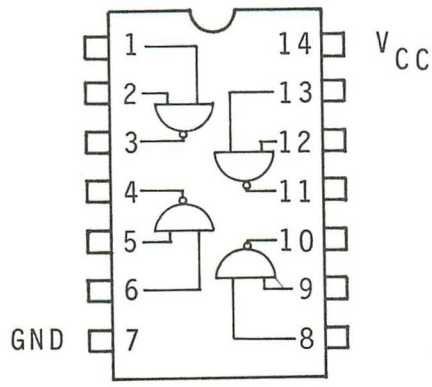


Wahrheitstabelle

Eingänge				Ausgänge	
Takt	Daten	Rücks.	Setzen	Q	$\overline{Q}$
C	D	R	S		
↗	Ø	Ø	Ø	Ø	1
↗	1	Ø	Ø	1	Ø
↘	X	Ø	Ø	Q	$\overline{Q}$
X	X	1	Ø	Ø	1
X	X	Ø	1	1	Ø
X	X	1	1	1	1

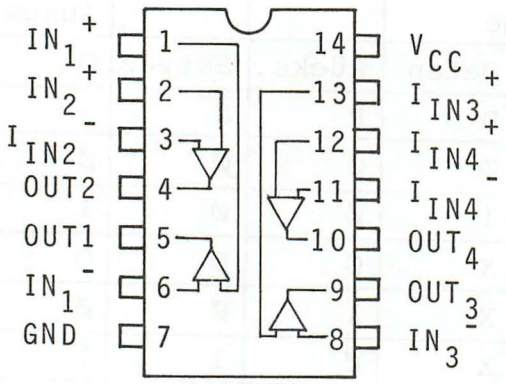
4.3.11.3 MC14011, Vierfach-NAND-Gatter

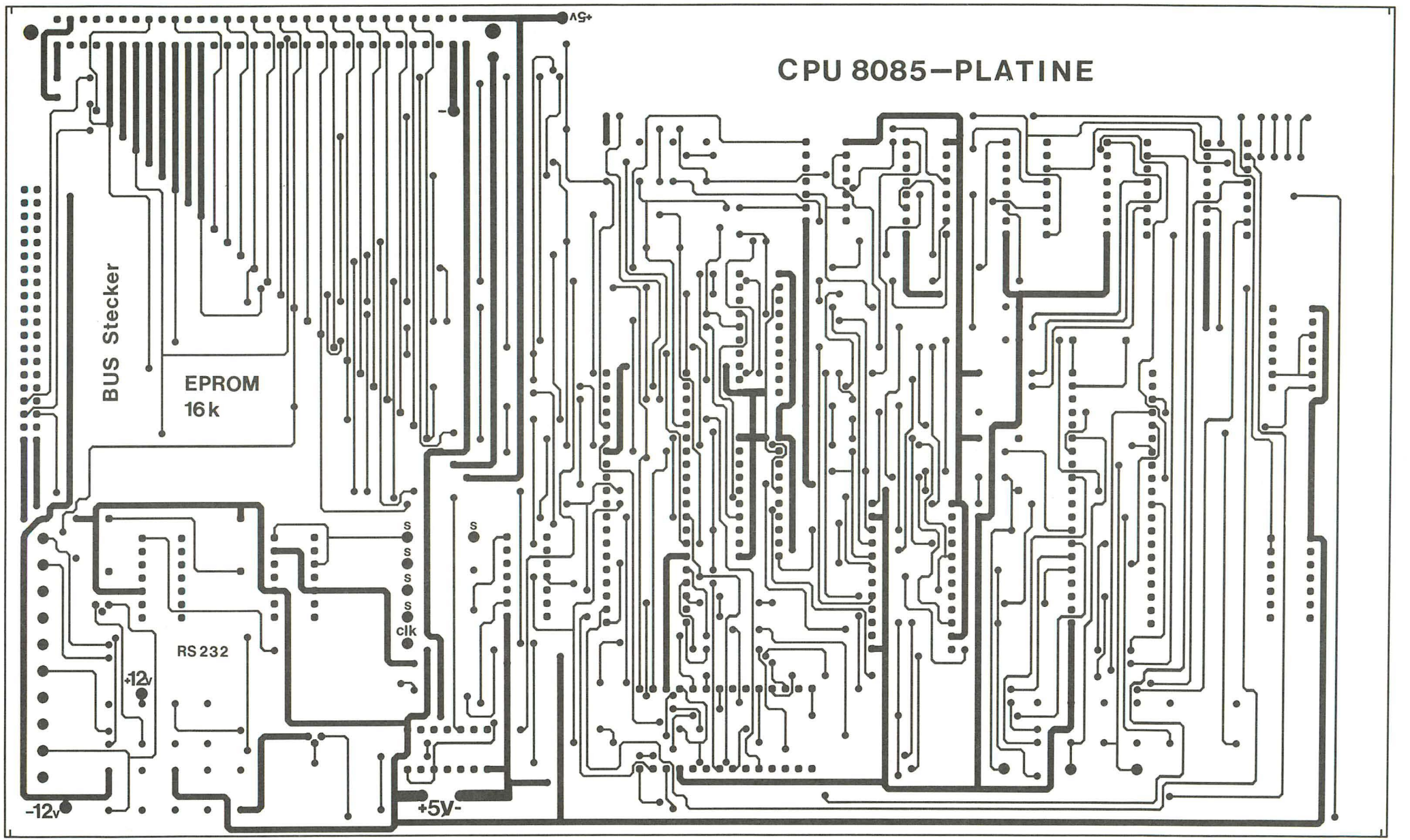
Anschlußbelegung



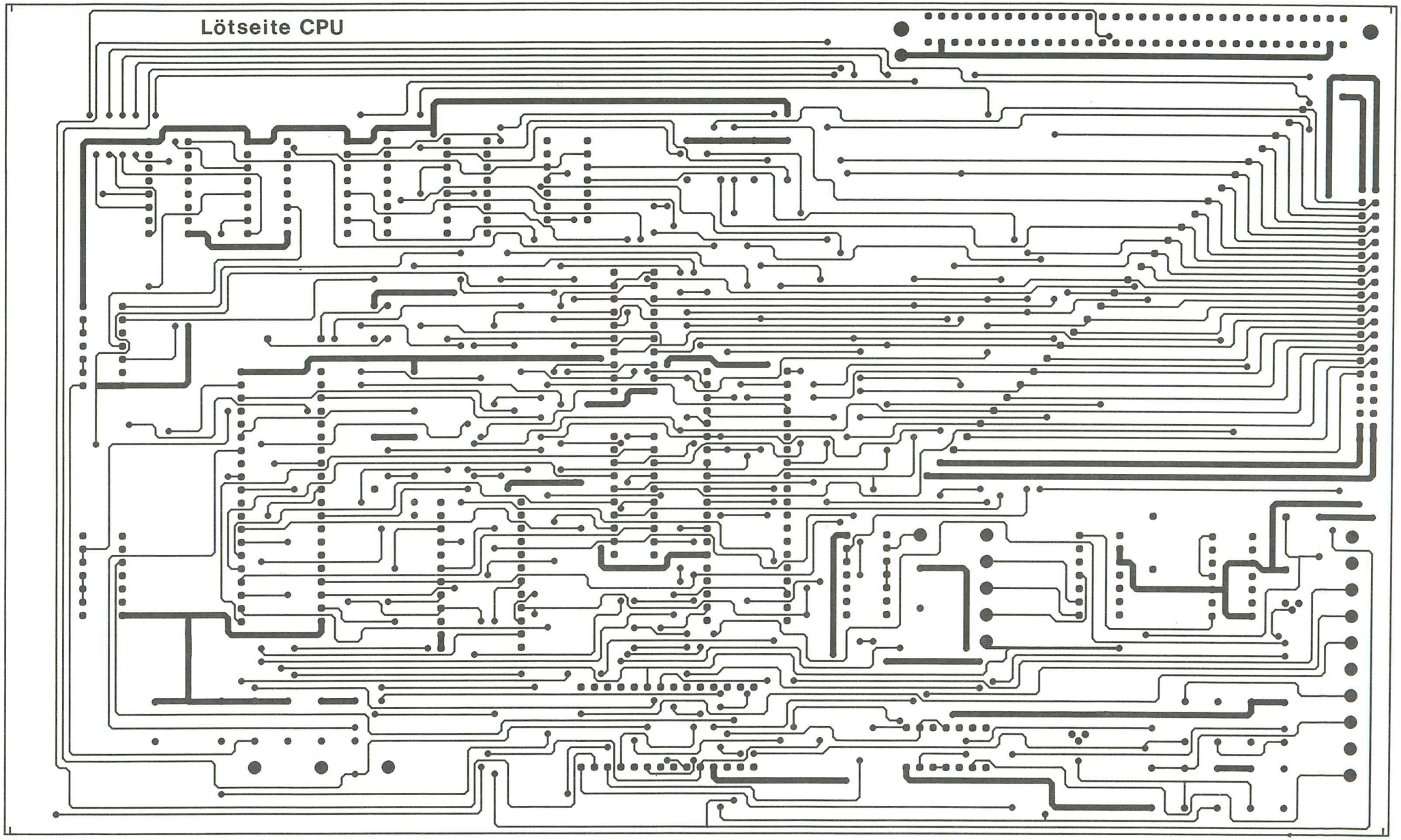
4.3.11.4 LM3900, Operationsverstärker/Puffer

Anschlußbelegung

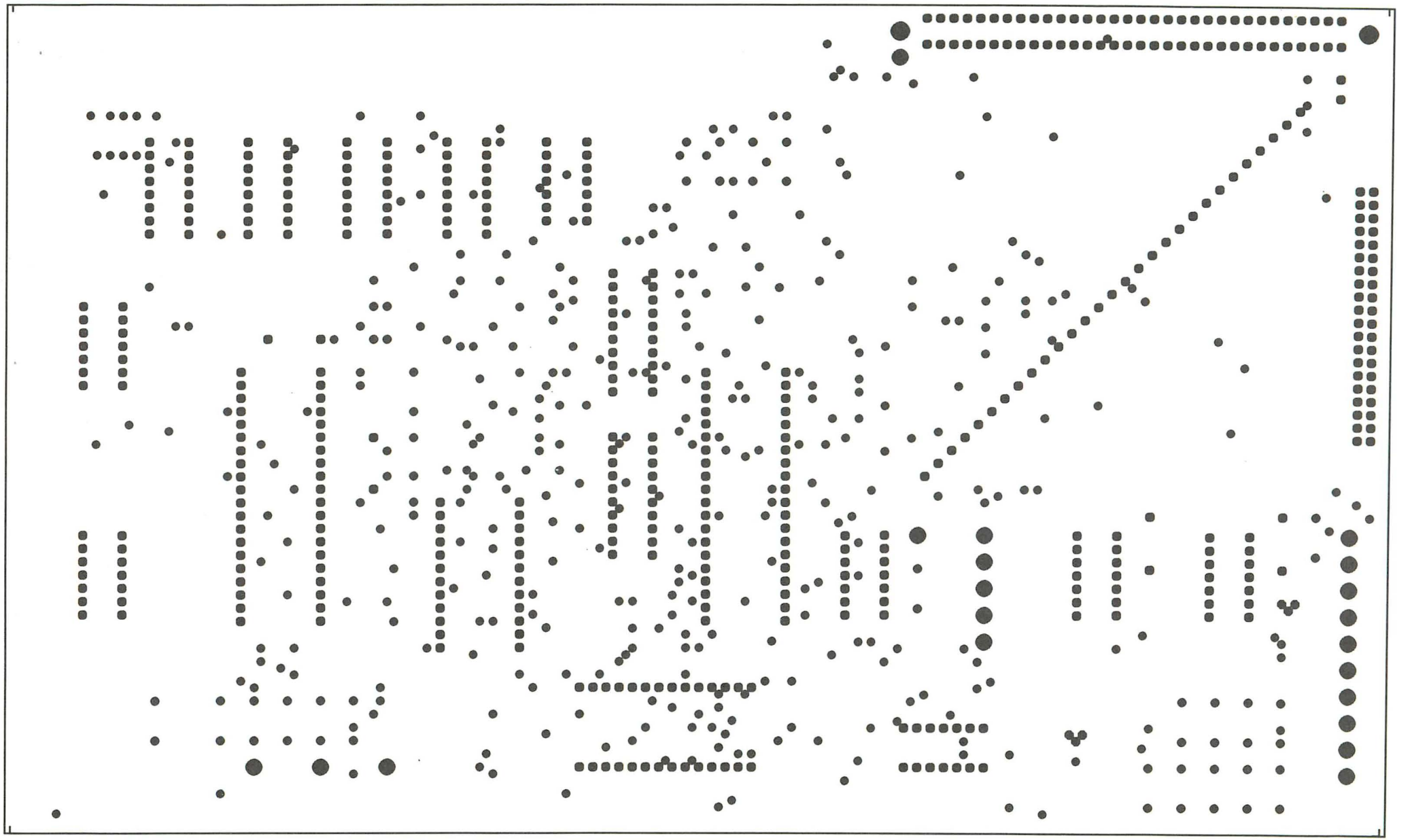




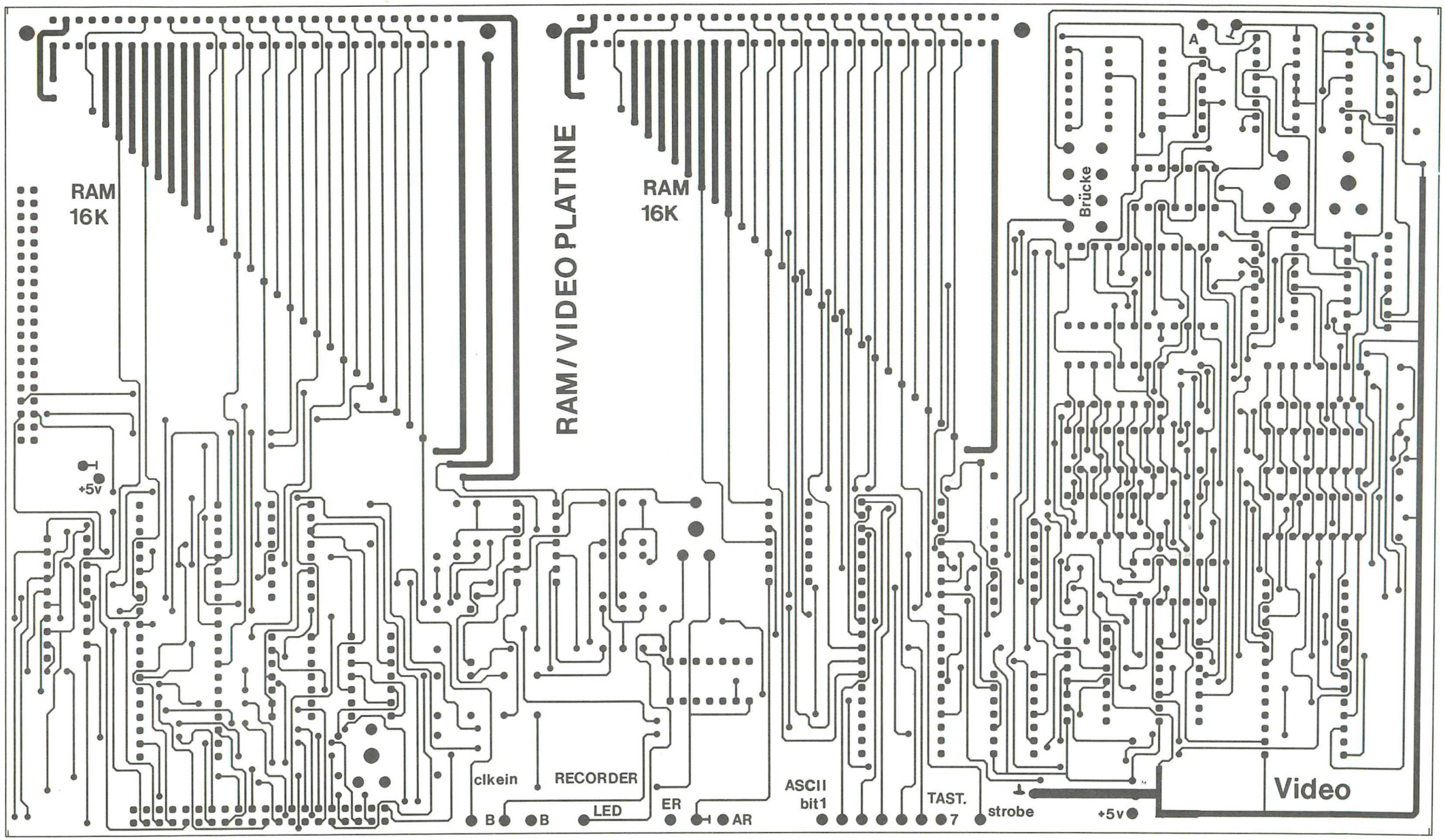
CALAY V01 25-MAY-82 DX1:VIDEO.PLT / BESTUECKUNGSSEITE



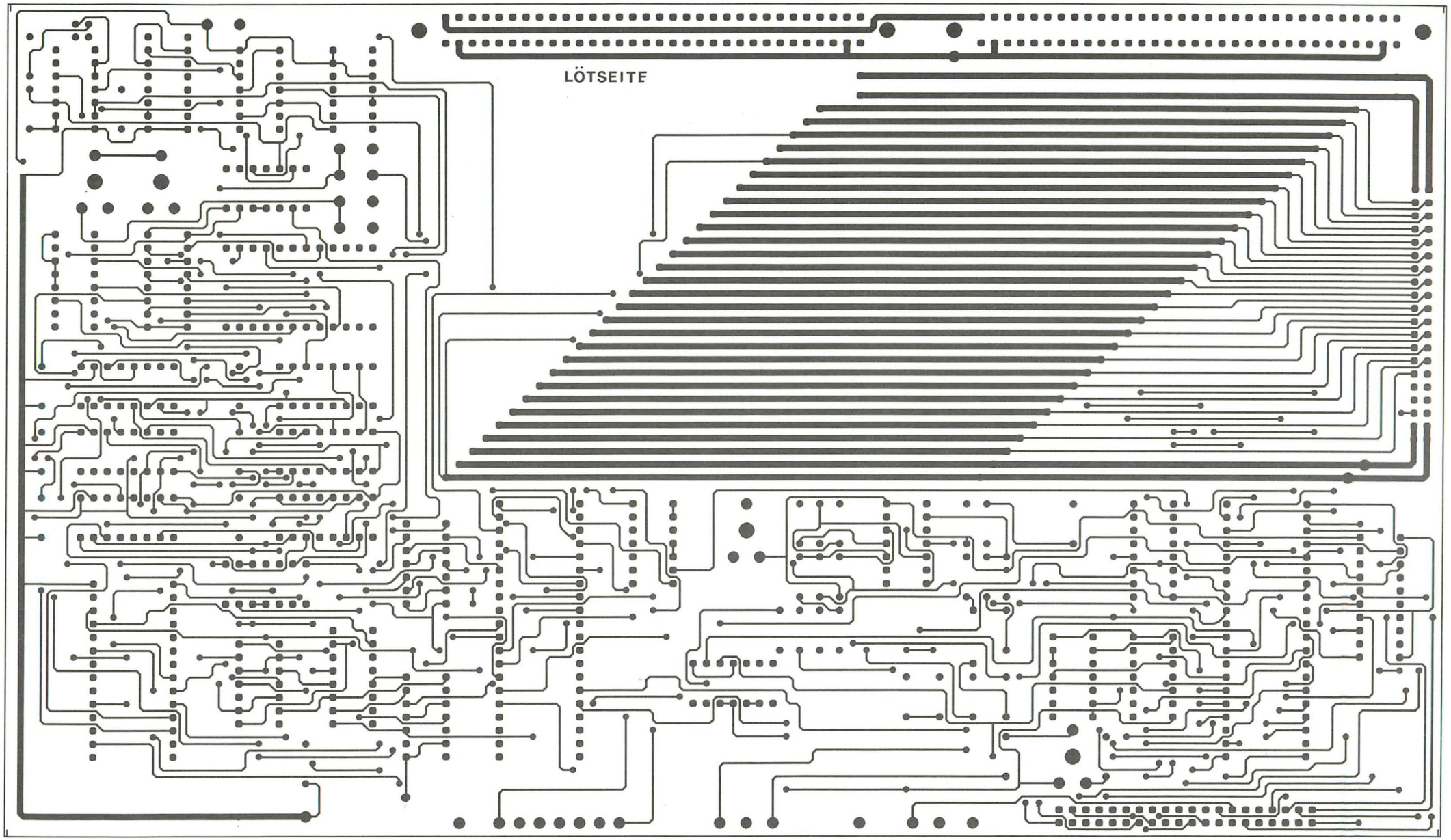
CALAY V01 25-MAY-82 DX1:VIDEO.PLT / LOETSEITE



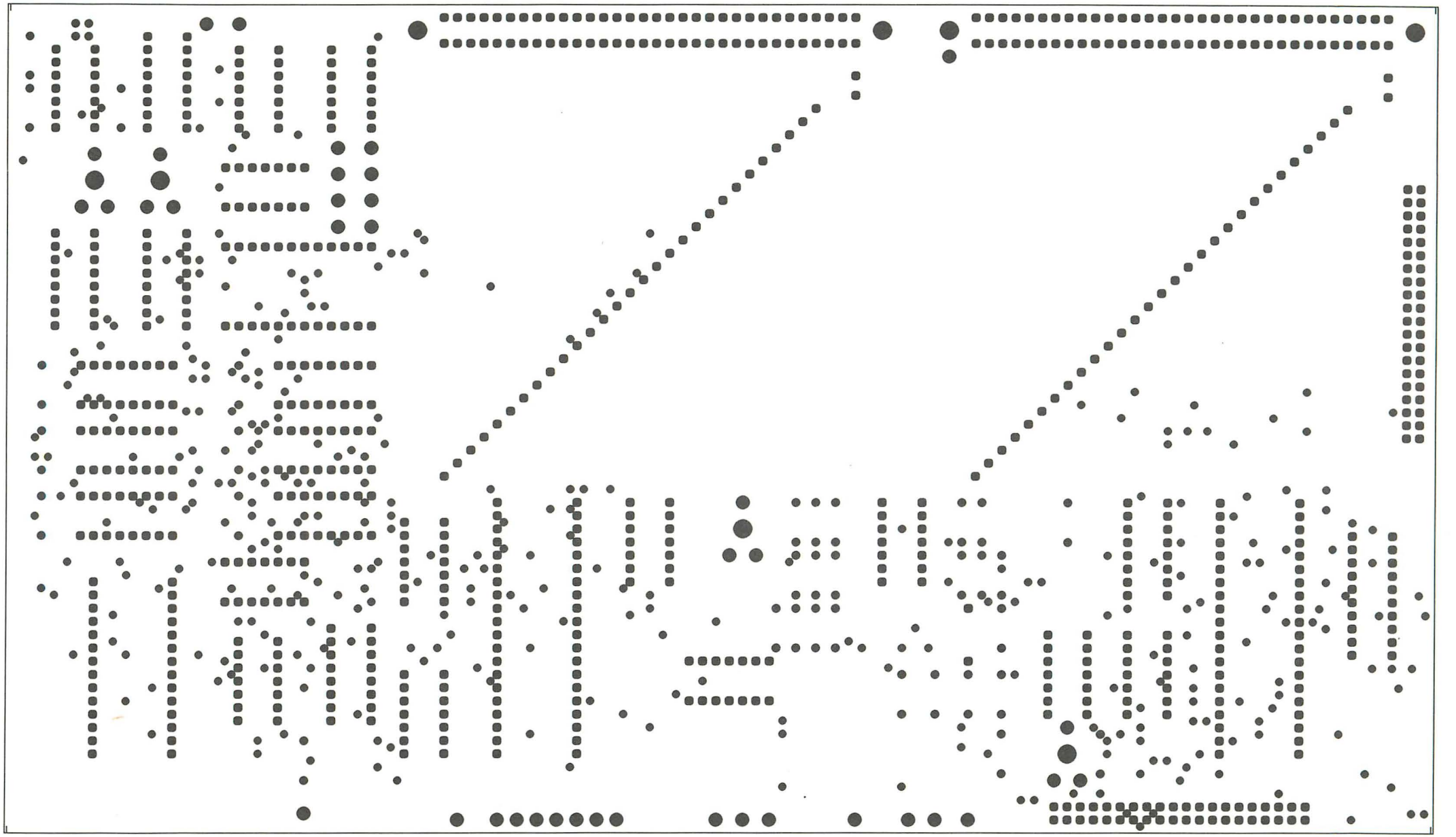
CALAY V01 25-MAY-82 DX1:VIDEO.PLT / LOETSTOPMASKE



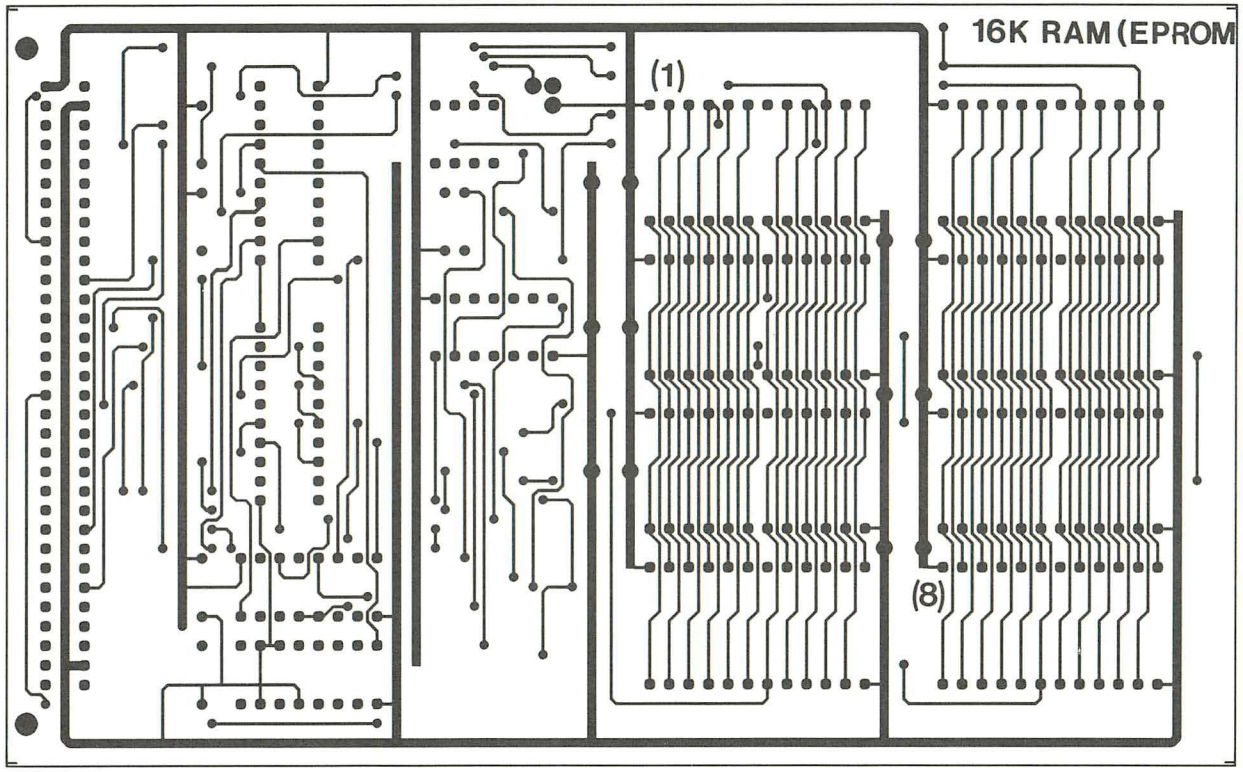
CALAY V01 18-JAN-83 DX1:SYSTEM.PLT / BESTUECKUNGSSEITE



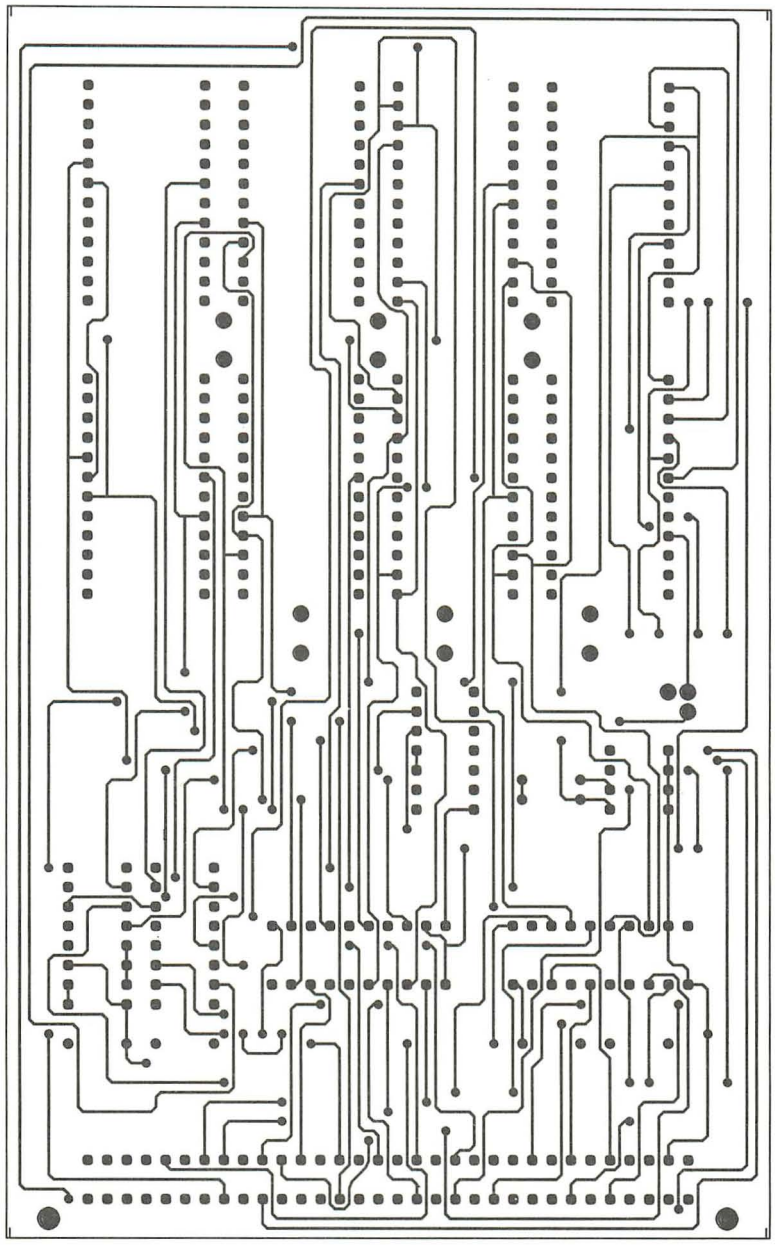
CALAY V01 18-JAN-83 DX1:SYSTEM.PLT / LOETSEITE

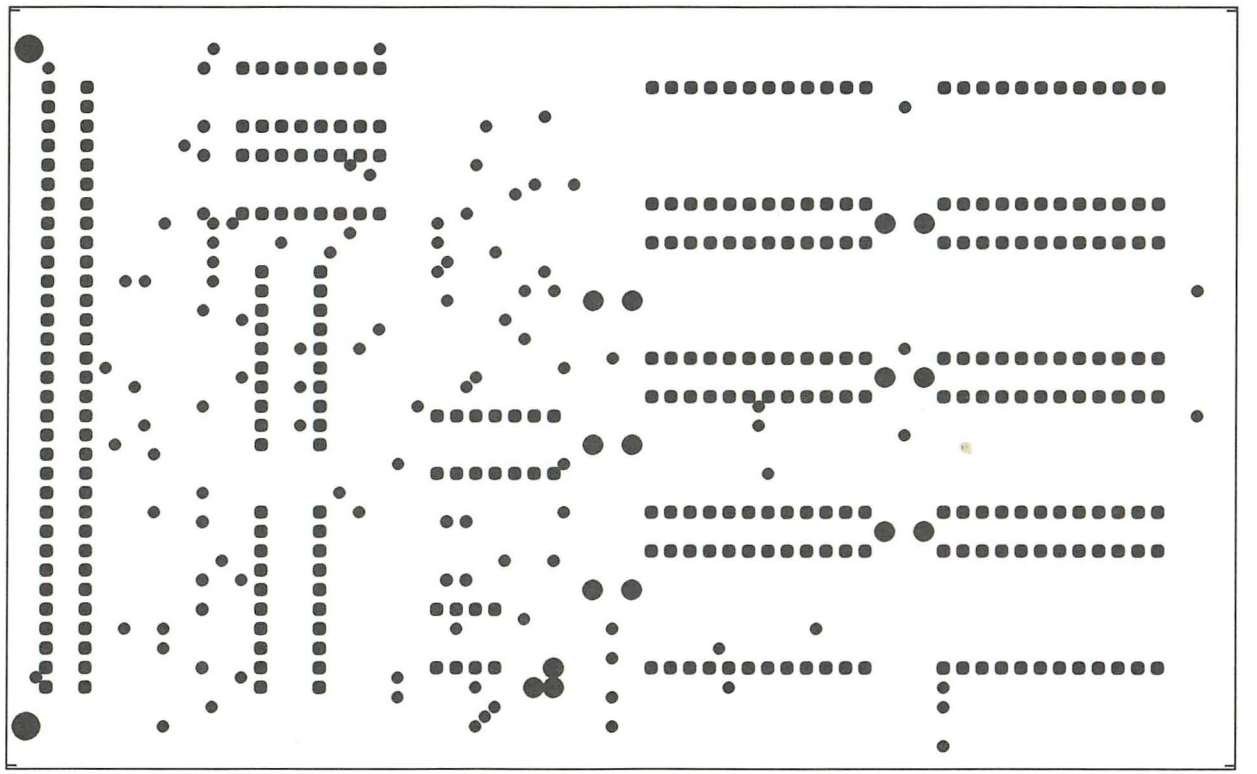


CALAY V01 18-JAN-83 DX1:SYSTEM.PLT / LOETSTOPMASKE



CALAY V01 12-FEB-83 DX1:CRAM16.PLT / BESTUECKUNGSSEITE





CALAY V01 12-FEB-83 DX1:CRAM16.PLT / LOETSTOPMASKE

6. BESCHREIBUNG DER SOFTWARE

In den bisherigen Abschnitten stand die Schaltungstechnik, die sogenannte HARDWARE des CHIP-Computers im Vordergrund.
Da auch dieser Microcomputer schließlich nur das ausführt, was ihm zuvor aufgetragen worden ist, muß man ihm mitteilen was er tun soll. Alle Informationen, die nun im System abgespeichert und verarbeitet werden, bestehen letztendlich aus einer Folge von gesetzten und nicht gesetzten Bits. Im CHIP-Computer ist die Grundlage der Informationsdarstellung ein 8-BIT breites BITMUSTER, das vom Microprozessor 8085 verarbeitet wird. Es lassen sich somit also $2^8=256$ verschiedene Zustände darstellen, wobei es vom Anwender abhängt, welche Art von Information er diesen möglichen Zuständen zuordnen will.
Das Computersystem kann also nur solche Programme ausführen, deren Befehle in einer BINÄR-CODIERTER Form vorliegen, dem sogenannten MASCHINENCODE.
Da es äußerst mühsam ist, ein PROGRAMM in diesem Code zu erstellen, wurden Programmiersprachen entwickelt, eine Art der Programmiersprache ist die sogenannte ASSEMBLER-Sprache, in der jedem Computer-Befehl in eindeutiger Weise ein bestimmtes SYMBOL zugeordnet ist. Jeder Befehl der ASSEMBLER-Sprache wird mit Hilfe eines Übersetzungsprogramms anschließend in den entsprechenden MASCHINEN-CODE übersetzt.
Mit dem vorhandenen ASSEMBLER hat der Entwickler nun ein Mittel in der Hand, mit dem er Programme schreiben kann, ohne sich dabei um das Format der MASCHINENBEFEHLE und die physikalische Adressierung des Programms zu kümmern.
Für die Eingabe des ASSEMBLER-Quellprogramms wird ein Dienstprogramm aufgerufen, der sog. EDITOR. Er ist eines der wichtigsten SOFTWARE-Hilfsmittel dieses Systems und dient zur Eingabe und Korrektur von Programmen.
In den folgenden Abschnitten wird nun auf das Betriebssystem des CHIP-Computers eingegangen, wobei EDITOR, ASSEMBLER u. die notwendigen TESTHILFEN eingehend beschrieben werden.

6.1 ÜBERSICHT DER BEFEHLSEINGABE

Die BEFEHLSEINGABE erfolgt über die sogenannte STATUS: Linie, mit der sich der CHIP-Computer nach Einschalten der Stromversorgung meldet. Dabei sind folgende EINGABEN an das System möglich:

6.1.1 STATUS:EDIT NEW
Aufruf zum Editieren eines neuen FILES(Datei).

6.1.2 STATUS:EDIT OLD
Aufruf zum Editieren eines im internen RAM-Speicher bereits vorhandenen FILES.

6.1.3 STATUS:ASSEMBLE
Aufruf des ASSEMBLERS, der das QUELLPROGRAMM in den MASCHINENCODE übersetzt.

6.1.4 STATUS:DISPLAY MEMORY 7000H
Auflisten der Speicherinhalte von ADRESSE 7000H.

6.1.5 STATUS:MODIFY MEMORY 7000H TO FFH
Setzen eines Speicherplatzes auf den Wert FF.

6.1.6 STATUS:MODIFY MEMORY 7000H TO FFH(05)
Modifizieren bis zu (99) Speicherplätzen.

6.1.7 STATUS:RUN FROM 8000H
ABSOLUTE Startadresse setzen.

6.1.8 STATUS:RUN FROM OTTO
SYMBOLISCHE Startadresse setzen.

6.1.9 STATUS:STEP FROM 8000H
Startadresse für EINZELSCHRITT setzen.

6.1.10 STATUS:STEP FROM OTTO
SYMBOLISCHE Startadresse für Einzelschritt.

6.1.11 STATUS:MODIFY REGISTER A TO FFH
Setzen der Registerinhalte der CPU 8085.

6.1.12 STATUS:RUN FROM 8000H UNTIL 8030H
Setzen einer START- und einer STOPADRESSE.

6.1.13 STATUS:RUN FROM OTTO UNTI EMIL
Setzen einer SYMBOLISCHEN START-u.STOPADRESSE.

6.1.14 STATUS:PRINT
Ausdrucken des Quellprogramms auf den Drucker.

6.1.15 STATUS:STORE SOURCE
Abspeichern des QUELLPROGRAMMS auf Recorder.

6.1.16 STATUS:LOAD SOURCE
Laden eines QUELLPROGRAMMS vom Recorder.

6.1.17 STATUS:READ (Systemerweiterung)
Auflisten von EPROM 2716 Daten auf Bildschirm.

6.1.18 STATUS:EPROM (Systemerweiterung)
Programmieren eines 2716 EPROMs.

6.1.19 STATUS:PROMCHECK (Systemerweiterung)
Testen eines 2716 EPROMs auf Programmierfähigkeit.

6.1.20 STATUS:VERIFY (Systemerweiterung)
Verifizieren der EPROM-Daten.

6.1.21 STATUS:LIST (Systemerweiterung)
Ausdrucken der EPROM-Daten auf Drucker.

Alle Befehle werden mit der [CR] TASTE (RETURN) abgeschlossen.

Eine Korrektur falsch eingegebener Zeichen ist mit den CURSOR-Steuertasten möglich. In der STATUS: Linie sind dabei folgende CURSOR-Steuerungen möglich (Keine Wiederholsteuerung):

---➨ CURSOR RECHTS
⬅--- CURSOR LINKS

Die für das System nicht erkennbaren BEFEHLE werden über die STATUS: Linie folgendermaßen angezeigt:

STATUS: EINGABEFEHLER!

Um nun eine neue Befehlseingabe zu starten, muß vorher die STATUS: Linie gelöscht werden. [LEER] TASTE betätigen.

STATUS:

6.2 EDITIEREN EINES NEUEN PROGRAMMS

STATUS:EDIT NEW

Bei dieser Befehlseingabe wird der gesamte EDITIER-Speicher, RAM 4000H bis RAM 9000H gelöscht, d.h. eine vorher geschriebene DATEI (FILE) ist nicht mehr vorhanden.
Auf dem Bildschirm ergibt sich folgende Darstellung:

```
++++++++++++++++++++++++++++++++++++++++
      EDITIEREN EINES NEUEN FILES
      NAME:_                DATUM:
++++++++++++++++++++++++++++++++++++++++
```

FILE-Namen und Datum eintragen und mit [CR] TASTE abschließen, das System meldet sich dann anschließend mit NEW in der ersten ZEILE.

```
++++++++++++++++++++++++++++++++++++++++
       EDITIEREN EINES NEUEN FILES
       NAME:CHIP-COMPUTER    DATUM: 1983
++++++++++++++++++++++++++++++++++++++++
NEW_
```

Nun kann mit dem eigentlichen Schreiben des Quellprogramms begonnen werden, dabei stellen die CURSOR-Tasten eine hilfreiche Funktion dar, um ev. Korrekturen an einem Programm vornehmen zu können.

[►]---- CURSOR RECHTS
[◄]---- CURSOR LINKS

[▲]---- CURSOR OBEN

[▼]---- CURSOR UNTEN

Die CURSOR-Tasten OBEN u. UNTEN sind aber nur dann in Betrieb, wenn eine geschriebene ZEILE mit einem [CR] Tastendruck abgeschlossen wurde.

```
++++++++++++++++++++++++++++++++++++++++
       EDITIEREN EINES NEUEN FILES
       NAME:CHIP-COMPUTER    DATUM  1983
++++++++++++++++++++++++++++++++++++++++
NEW   DIES IST EIN_ CR
NEW   8085 MICROCOMPUTERSYSTEM_   CR
NEW
```

Weiterhin muß man unbedingt beachten, daß alle <u>rechts</u> vom CURSOR stehenden Zeichen durch ein [CR] Tastendruck gelöscht werden. Insgesamt lassen sich auf einer Bildschirmseite 16 ZEILEN mit 63 Zeichen (einschl. NEW) schreiben.

Steht der CURSOR nun auf der letzten ZEILE im Bildschirm, die man, wie beschrieben, mit der [CR] Taste abschließt, so wird die 1. ZEILE des Programms aus der Anzeige ausgeblendet(SCROLLMODUS). Auf dem Bildschirm ergibt sich dann folgende Darstellung:

```
2.              EDITIEREN EINES NEUEN FILES
3.              NAME:CHIP-COMPUTER    DATUM:  1983
4.      ++++++++++++++++++++++++++++++++++++++++++++
5.      NEW   DIES IST EIN
6.      NEW   8085 MICROCOMPUTERSYSTEM
7.      NEW   HIER SOLLTEN EINIGE
8.      NEW   ZEILEN GESCHRIEBEN WERDEN,
9.      NEW   UM DEN EDITOR KENNENZULERNEN.
10.     NEW   ER IST LEICHT ZU BEDIENEN,
11.     NEW   NUR
12.     NEW   MUSS
13.     NEW   MAN
14.     NEW   EINIGE
15.     NEW   EIGENSCHAFTEN DIESES
16.     NEW   BILDSCHIRM-ORIENTIERTEN EDITORS KENNENLERNEN.
17.     NEW   DIE ERSTE ZEILE WURDE HERAUSGESCROLLT.
```

6.2.1 FUNKTION EINIGER STEUERTASTEN

6.2.1.1 Die Steuerfunktion [CONTROL] [S] (13H ASCII)

Die STEUERTASTE [CONTROL] [S] ermöglicht es zum Programmanfang zurückzukehren. Dabei werden die ersten 16 ZEILEN des Quellprogramms im Bildschirm aufgelistet.
CURSOR-Stand hinter NEW, letzte Eingabe mit [CR] abgeschlossen.
Folgende Beispiele sollen die Steuerfunktion [CONTROL] [S] noch näher erläutern:

1. BEISPIEL:

```
++++++++++++++++++++++++++++++++++++++++++++
        EDITIEREN EINES NEUEN FILES
        NAME:CHIP-COMPUTER    DATUM:  1983
++++++++++++++++++++++++++++++++++++++++++++
NEW       ORG 8000H
NEW       MOV A,B
NEW+++ FOLGENDE BEFEHLE SOLLEN NUR DIE
NEW+++ WIRKUNG DER STEUERTASTEN ZEIGEN!
NEW       MVI A,80H
NEW       LXI H,OTTO
NEW OTTO EQU 9000H
NEW       END
NEW_
```

Eine Betätigung der [CONTROL] [S] TASTE ist in diesem Beispiel ohne Wirkung, das Programm ist bereits in der ersten Seite beendet.

2. BEISPIEL:

```
25.       NEW     MOV A,C
26.       NEW     ADI 01H
27.       NEW     SUB A
28.       NEW     LXI D,7000H
29.       NEW     LXI B,7400H
30.       NEW     END
31.       NEW_
```

In diesem Beispiel ist das Programm in der 30. ZEILE beendet, jetzt würde eine [CONTROL] [S] Anweisung die ersten 16 ZEILEN des Programms nochmals auf den Bildschirm geben.

TASTENFOLGE: [CONTROL] [S] betätigen, dann [CR] drücken.

(Erst CONTROL drücken u.halten, dann S Taste betätigen)

6.2.1.2 Die Steuerfunktion [CONTROL] [P] (10H ASCII)

Mit dieser Steuertaste können jeweils 16 ZEILEN eines editierten Programms auf dem Bildschirm dargestellt werden. CURSOR-Stand hinter NEW.

BEISPIEL: Ein geschriebenes Programm umfaßt 50 Zeilen und soll von der ersten SEITE (Zeile 1 - Zeile 16) an nochmals überprüft werden:

```
44.     NEW OTTO MOV A,B
45.     NEW      MOV B,C
46.     NEW      ADI 01H
47.     NEW      JC OTTO
48.     NEW      LXI D,7000H
49.     NEW      JMP 8300H
50.     NEW      END
51.     NEW_
```

[CONTROL] [S] betätigen und somit zur 1. SEITE des Programms gehen:

```
        ++++++++++++++++++++++++++++++++++++++++++++++++++
                EDITIEREN EINES NEUEN FILES
                NAME:CHIP-COMPUTER    DATUM  1983
        ++++++++++++++++++++++++++++++++++++++++++++++++++
5.      NEW      ORG 8000H
6.      NEW      MOV A,B
         "
16.     NEW      MOV B,C_
```

[CR] Taste betätigen und anschließend mit einer [CONTROL] [P] Anweisung jeweils die nächsten 16 ZEILEN des Programms auf den Bildschirm darstellen.

6.2.1.3 Die Steuerfunktion CONTROL E (05H ASCII)

Die CONTROL E Steuertaste ermöglicht es in einer Bildschirmseite eine neue ZEILE einzufügen. In der ersten und letzten ZEILE ist ein Einfügen aber nicht möglich.

BEISPIEL: In der ersten SEITE eines Programms soll zwischen der 8. und 9. ZEILE ein zusätzlicher Befehl eingefügt werden.

```
        ++++++++++++++++++++++++++++++++++++++++++++++
                EDITIEREN EINES NEUEN FILES
                NAME:CHIP-COMPUTER    DATUM:  1983
        +++++++++++++++++++++++++++++++++++++++++++++++
5.      NEW      ORG 8200H
6.      NEW      MOV A,B
7.      NEW      MOV C,B
8.      NEW_     MOV A,D; CURSORSTAND ZUM EINFUEGEN
9.      NEW      MOV A,E
10.     NEW      MOV A,H
11.     NEW      MOV A,L
12.     NEW_
```

Der CURSOR wird nun in diesem Fall mit der Steuertaste CURSOR OBEN in die 8. ZEILE gesetzt, eine anschließende CONTROL E Anweisung eröffnet dann eine neue ZEILE:

```
                EDITIEREN EINES NEUEN FILES
                NAME:CHIP-COMPUTER    DATUM:  1983
        ++++++++++++++++++++++++++++++++++++++++++++++
5.      NEW      ORG 8200H
6.      NEW      MOV A,B
7.      NEW      MOV C,B
8.      NEW      MOV A,D
        NEW_
9.      NEW      MOV A,E
10.     NEW      MOV A,H
11.     NEW      MOV A,L
```

CR Taste betätigen und mit CURSOR OBEN in die ZEILE gehen.

6.2.1.4 Die Steuerfunktion [ESC] (1BH ASCII)

Das Aussteigen aus dem EDITOR wird mit einer ESC Anweisung vorgenommen. Diese ist aber nur dann funktionsfähig, wenn dem System vorher ein [CONTROL] [S] Befehl mitgeteilt worden ist.

BEISPIEL: Das Programm ist bereits in der ersten Bildschirmseite zu Ende.

```
1.      ++++++++++++++++++++++++++++++++++++++++++++++
2.              EDITIEREN EINES NEUEN FILES
3.              NAME:CHIP-COMPUTER    DATUM:  1983
4.      ++++++++++++++++++++++++++++++++++++++++++++++
5.      NEW       ORG 9000H
6.      NEW       MOV A,B
7.      NEW       MOV A,C
8.      NEW       MOV A,D
9.      NEW       END
10.     NEW_
```

Bevor jetzt aus dem EDITIER-Mode ausgestiegen werden kann, müssen die ersten vier ZEILEN aus dem Bildschirm herausgescrollt werden. Dies läßt sich mit entsprechenden [CR] Tastenfolgen realisieren:

```
NEW       ORG 9000H
NEW       MOV A,B
NEW       MOV A,C
NEW       MOV A,D
NEW       END
NEW
 "
NEW_
```

Wird nun eine [CONTROL] [S] Taste betätigt, listet das System die erste Bildschirmseite nochmals auf. Mit einer unmittelbar darauf folgenden [ESC] Anweisung beendet der CHIP-Computer den EDITIER-Mode und meldet sich mit der bekannten STATUS: Linie. Besteht ein Programm aus mehreren SEITEN , so gibt es zum

Aussteigen aus dem EDITOR noch folgendes zu sagen:

1. Befindet man sich auf eine der folgenden SEITEN, letzte ZEILE mit [CR] abgeschlossen, so kann man mit:
   [CONTROL] [S] und
   [ESC] die Erstellung des QUELLPROGRAMMS beenden.
2. Befindet man sich in der 1. SEITE und hat eine EINFÜG-Anweisung vorgenommen, so bedeutet dies beim Umfang mehrere SEITEN, daß zuerst ein [CONTROL] [P] Befehl übermittelt werden muß. Ansonsten Reihenfolge der Anweisungen wie 1.

6.2.1.5 Die Steuerung einer TABULATOR-Funktion (OFH ASCII)

Wird eine Tastatur Typ CHERRY G80-0247 eingesetzt, so kann mit Hilfe der [HOME] Taste eine TABULATOR-Funktion realisiert werden, d.h. der CURSOR überspringt das sog. MARKENFELD (siehe auch 6.4 ASSEMBLER).

6.3 EDITIEREN EINES VORHANDENEN PROGRAMMS

STATUS:EDIT OLD

Nach dieser Anweisung werden, je nach Umfang der Datei, die ersten 16 ZEILEN eines im internen RAM-Speicher abgelegten Programms auf den Bildschirm gebracht.
Im Unterschied zum EDIT NEW sind jetzt alle Zeilen durchgehend nummeriert.
BEISPIEL:

```
++++++++++++++++++++++++++++++++++++++++++++++
        EDITIEREN EINES NEUEN FILES
        NAME:CHIP-COMPUTER    DATUM:  1983
++++++++++++++++++++++++++++++++++++++++++++++
001      ORG 9000H
002      MOV A,B
003      MOV A,C
usw.
```

[CR] Taste betätigen, sonstige ANWEISUNGEN wie in 6.2 beschrieben.

STATUS:KEIN SOURCE FILE VORHANDEN!

Meldung des Systems, daß sich kein QUELLPROGRAMM im internen RAM-Speicher befindet. [LEER] Taste betätigen, damit kann die STATUS: Linie wieder beschrieben werden:

STATUS:

6.4 AUFRUF DES ASSEMBLERS

STATUS:ASSEMBLE

Mit diesem Kommando wird das Übersetzungsprogramm aufgerufen,daß das editierte QUELLPROGRAMM in den MASCHINENCODE übersetzen soll. Befehle und Anweisungen der ASSEMBLER-Sprache werden in eine Anzahl von Feldern aufgeteilt. Der im CHIP-COMPUTER vorhandene Übersetzer benötigt unbedingt folgende Einteilung:

MARKENFELD	OPERATIONSCODE	OPERANDENFELD
4 Zeichen	(siehe MNEMONIKS	8085/8080)

BEISPIEL:

```
NEW_OTTO_ MOV_A,B
NEW       LXI_H,8300H
NEW       MVI_A,80H
NEW       OUT_07H
NEW       XCHG
NEW       JMP_OTTO;REIHENFOLGE DER BEFEHLE KEINE BEDEUTUNG!
```

6.4.1 MARKEN (LABELS)

Das MARKENFELD ist das erste Feld eines Befehls in der 8085-ASSEMBLER-Sprache und muß nicht unbedingt beschrieben sein. Mit der Belegung des MARKENFELDES hat aber der Anwender die Möglichkeit nicht nur symbolische Adressen zu setzen, sondern

auch innerhalb eines Programms zugewiesene START-u.STOP-Adressen einzufügen. Dies erleichtert unter Umständen den durchzuführenden SOFTWARE-Test.
Im QUELLPROGRAMM gesetzte MARKEN müssen immer vier ZEICHEN lang sein, das erste Zeichen ist immer ein Buchstabe. Alle MARKEN werden während des Übersetzungslaufes in absolute ADRESSEN umgewandelt und in einem sogenannten SYMBOLSPEICHER abgelegt.
Ein LABEL gleicher Zeichenfolge darf nur einmal erscheinen,"END" als MARKE ist unzulässig.

BEISPIEL:

```
NEW OTTO MOV A,B
NEW EMIL MVI B,90H
NEW      LDAX B
NEW      ADI 00H
NEW      JZ OTTO
NEW      JMP EMIL
NEW
```

6.4.2 OPERATIONSCODE

Zwischen OPERATIONSCODE und OPERANDENFELD steht ein Leerzeichen.

BEISPIEL:

```
NEW      MOV_B,D
NEW      LXI_H,8300H
NEW      MOV_A,M
NEW      CALL_ADR1
NEW      JMP_OTTO
NEW
```

6.4.3 KOMMENTARE

Um ein Programm übersichtlich und detailliert schreiben zu können, besteht die Möglichkeit Kommentarfelder in das Programm einzubinden.

BEISPIEL:

```
NEW      ORG 9000H          (Kein Kommentar möglich!)
NEW      MOV B,H;INHALT VON H NACH B
NEW+++  DIES IST EIN TESTPROGRAMM  ++++
NEW      LXI H,8000H;HL ALS ADRESSREGISTER
NEW      INX H
NEW      MOV A,M
NEW      END                (Kein Kommentar möglich!)
```

Kommentare können,wie oben gezeigt, auf zwei verschiedene Arten in ein Programm aufgenommen werden:

1. Nach einer Befehlsanweisung durch Setzen eines Semikolons (MOV B,H; INHALT VON H NACH B).

2. Durch mindestens zwei ++ (STERN) kann eine ganze Kommentarzeile geschrieben werden. Nach der ORG-Anweisung darf aber keine Kommentarzeile folgen. Zeichenfolge "END" innerhalb eines Kommentares nicht erlaubt.

6.4.4 ASSEMBLER MNEMONIKS

Die hauptsächliche Ausgabe des ASSEMBLERS ist die Übersetzung von sog. MNEMONIKS in ihren äquivalenten OPERATIONSCODE. Dabei muß aber eine bestimmte Reihenfolge der ANWEISUNGEN undedingt eingehalten werden:

1. Die erste ANWEISUNG an das Übersetzungsprogramm <u>ist immer</u> ein ORG (Origin=Ursprung)-Hinweis, der dem ASSEMBLER mitteilt, in welchem Speicherberich er den MASCHINENCODE zu legen hat.

   NEW ORG 9000H bedeutet:

   Lege das Programm von Speicheradresse 9000H an. Hierbei ist es sinnvoll alle zu übersetzende Programme von der Adresse 8000H anzulegen, da der erste RAM-BLOCK für das Quellprogramm von 4000H an belegt ist.

Innerhalb eines QUELLPROGRAMMS können selbstverständlich auch mehrere ORG-Anweisungen erscheinen.

BEISPIEL:

```
NEW      ORG 8000H
NEW      LXI H,8330H
NEW      INX H
NEW      MOV B,M
NEW      INX H
NEW      MOV C,M
NEW      ORG 8300H
NEW      DB 01H;SETZE (8300H)=01
NEW      DB 02H;      (8301H)=02
NEW      DB 03H
NEW      END
```

Die letzte ANWEISUNG an den ASSEMBLER ist <u>immer</u> ein END, der dem Übersetzungsprogramm mitteilt, hier ist das QUELLPROGRAMM zu Ende.

Alle KONSTANTEN (8-BIT u. 16-BIT) werden mit H (H=HEX) abgeschlossen, wobei noch darauf hingewiesen werden muß, daß die 16-BIT-ADRESSEN-Darstellung auch in SYMBOLISCHER FORM vorgenommen werden kann und mit einer am Ende des Programms zugewiesenen EQU-Anweisung definiert wird.

BEISPIEL:

```
NEW      LDA 9000H
NEW      LDA OTTO
NEW      LDA ØAØØØH;          EINE ABSOLUTE ADRESSE AØØØH!
NEW OTTO EQU 83FFH
NEW      END
```

6.4.5 ASSEMBLER-ANWEISUNGEN(Pseudo-Operationen)

Manche der 8085/8080 ASSEMBLER-Befehle werden nicht direkt in MASCHINENSPRACHEN-Befehle übersetzt. Diese "BEFEHLE" stellen Anweisungen für den ASSEMBLER dar und haben mit dem eigentlichen BEFEHLSSATZ des o.a. Microprozessors nichts zu tun.

Die gebräuchlisten Pseudo-Operationen können auch im CHIP-COMPUTER verwendet werden.

BEISPIEL:

```
NEW      ORG 8300H
NEW      DB 01H;WEIST DER ADRESSE 8300H DEN WERT 01H ZU
NEW      DW 0203H; (8301)=02H (8302)=03H
NEW      ORG 8200H
NEW      LXI H,OTTO
NEW      LXI D,EMIL
NEW        "
  "
NEW OTTO EQU 9000H
NEW EMIL EQU 9004H
NEW      END
```

6.4.5.1 STATUS:KEIN SOURCE FILE VORHANDEN!

Die erste Zeile eines QUELLPROGRAMMS muß mit einer ORG-Anweisung beginnen, die letzte Zeile beinhaltet immer den END-Befehl! Werden diese beiden Anweisungen nicht eingehalten, so meldet sich der CHIP-COMPUTER mit der o.a. STATUS:__Linie, [LEER] Taste betätigen.

6.4.6 STATUS:ASSEMBLER END NO ERRORS

Das QUELLPROGRAMM ist ohne Fehler übersetzt worden, um weitere KOMMANDOS eingeben zu können bitte [LEER] Taste betätigen.

6.4.7 STATUS:ASSEMBLER END ERRORS 2

Im Quellprogramm sind zwei SYNTAX-Fehler erkannt worden.

6.5 SPEICHER AUSLESEN UND BESCHREIBEN (Ausstieg: ESC Taste)

6.5.1 STATUS:DISPLAY MEMORY 8000H

Diese Befehlseingabe ermöglicht es, eine Darstellung von Speicherinhalten auf dem Bildschirm vorzunehmen.
Mit Betätigung der CR Taste werden dann jeweils die folgenden 15 Speicherinhalte dargestellt:

(8000H)=20
(8001H)=FF
(8002H)=00
usw.

6.5.2 STATUS:MODIFY MEMORY 8000H TO 40H

Mit dieser Anweisung wird der gewünschte Speicherplatz 8000H auf den Wert 40H gesetzt, Darstellung der Inhalte auf dem Bildschirm.

6.5.3 STATUS:MODIFY MEMORY 8000H TO 40H(10)

Von der Speicheradresse 8000H an, werden die folgenden Inhalte(10) auf den Wert 40H modifiziert.

6.6 KOMMANDOS ZUM SOFTWARETEST

6.6.1 STATUS:RUN FROM 8000H

Der Programmzähler der CPU 8085 wird mit dieser Anweisung auf eine gewünschte START-ADRESSE gesetzt und arbeitet dann das vom ASSEMBLER übersetzte Programm ab. Der CHIP-COMPUTER zeigt dies über seine STATUS:Linie folgendermaßen an:

STATUS:8085 RUNNING

Da die MASCHINENPROGRAMME in der Regel mit einem HLT (HALT-Befehl) begrenzt sind oder in eine Schleife laufen, kann somit nur

durch einen "hardwaremäßigen" RESET aus dem Programm ausgestiegen werden:

BREAK-Taste betätigen!

6.6.2 STATUS:RUN FROM OTTO

Befinden sich innerhalb des Quellprogramms verschiedene MARKEN, so kann auch von einer symbolischen Adresse aus gestartet werden, sonst wie 6.6.1.

6.6.3 STATUS:STEP FROM 8000H

Um MASCHINENPROGRAMME schrittweise austesten zu können, besteht die Möglichkeit dem CHIP-COMPUTER eine Step-Anweisung zu übergeben. Damit werden alle Registerinhalte der CPU 8085 auf dem Bildschirm dargestellt, wobei zugleich eine Rückübersetzung des MASCHINEN-CODES vorgenommen wird. (siehe auch 2.1 HARDWARE)

8085 OPCODE (B) (C) (D) (E) (H) (L) (A) CPAZS (SP) (PC)
+++ BEFORE+++ (bedeutet Inhalte der CPU-Register vor Start)

Bedeutung:

(B)=Inhalt von Register B	(SP)=Inhalt des STACKPOINTERS
(C)=Inhalt von Register C	(PC)=Stand des PROGRAMMZÄHLERS
(D)=Inhalt von Register D	C= CARRY-FLAG
(E)=Inhalt von Register E	P= PARITY-FLAG
(H)=Inhalt von Register H	A= AUXILIARY-FLAG
(L)=Inhalt von Register L	Z= ZERO-FLAG
(A)=Inhalt des AKKUMULATORS	S= SIGN-FLAG

Mit jedem folgenden [CR] Tastendruck werden die o.a. Registerinhalte und der 8085-OPCODE auf den Bildschirm gegeben.
Ausstieg: [ESC] Taste betätigen

6.6.4 STATUS:STEP FROM OTTO

Setzen einer symbolischen Startadresse für den STEP-Betrieb.

6.6.5 STATUS:MODIFY REGISTER A TO FFH

Mit diesem Kommando können bestimmte Register des 8085 vor einem STEP-Betrieb auf gewünschte Werte gesetzt werden.

6.6.6 STATUS:RUN FROM 8000H UNTIL 8050H

Um Programmabschnitte austesten zu können, wird dem CHIP-COMPUTER eine START-u.STOP-Adresse übergeben. Nach Erreichen dieser STOP-Adresse werden alle CPU 8085-Registerinhalte auf dem Bildschirm ausgegeben.Die weitere Abarbeitung des Programms erfolgt im STEP-Betrieb (siehe 6.6.3).

6.6.7 STATUS:RUN FROM 8000H UNTIL EMIL

Setzen einer absoluten Startadresse und einer symbolischen STOP-Adresse.(sonst wie 6.6.3).

6.6.8 STATUS:RUN FROM OTTO UNTIL EMIL

Setzen einer symbolischen START-u.STOP-Adresse.(sonst wie 6.6.3). Die Ausführung des MASCHINENPROGRAMMS wird bei den hier beschriebenen Unterbrechungsmöglichkeiten aber nur gestoppt, wenn die CPU auch tatsächlich die vorgeschriebene ADRESSE erhält, d.h. der Programmzähler des 8085 muß innerhalb des Programms mit dieser ADRESSE geladen werden.(sonst Ausstieg mit BREAK).

6.7 BEFEHLSEINGABE FÜR DRUCKER UND RECORDER

6.7.1 STATUS:PRINT

Ausdrucken des QUELLPROGRAMMS auf einen DRUCKER mit CENTRONICS-Kompatibler Schnittstelle (siehe auch HARDWARE 2.3.3).

6.7.2 STATUS:LOAD SOURCE

Mit dieser Befehlseingabe kann ein auf Band abgespeichertes

QUELLPROGRAMM in den CHIP-COMPUTER eingelesen werden. Zu diesem Zweck muß der Vorspann eines Programms über die LEUCHTDIODE beobachtet werden. Nach dem ein ca. 8s dauernder Blinkvorgang beendet ist, wird unmittelbar darauf die [CR] Taste betätigt, auf dem Bildschirm erscheint dann als Bestätigung:

STATUS:LOAD SOURCE+

Ist das Programm ohne Einlesefehler übermittelt worden, erscheint je nach Umfang der Datei die erste Seite auf dem Bildschirm. Anschließend [CR] Taste oder [ESC] Taste betätigen. Wurden im Gegenteil, während der Übertragung Einlesefehler vom System erkannt, so erscheint die STATUS:Linie in folgender Form:

STATUS:KEIN SOURCE FILE VORHANDEN

[LEER] Taste betätigen und Einlesevorgang wiederholen. Weiterhin muß man berücksichtigen, daß ein ev. im RAM-Speicher vorhandenes QUELLPROGRAMM durch die o.a. Anweisung überschrieben wird und somit nicht mehr vorhanden ist.

6.7.3 STATUS:STORE SOURCE+

Mit diesem Kommando wird ein QUELLPROGRAMM des CHIP-COMPUTERS auf dem Band abgespeichert. Bevor man diese Befehlseingabe mit der [CR] Taste quittiert, sollte der Recorder eingeschaltet werden und im Aufnahme-Mode arbeiten. Nach beendeter Übertragung erscheint:

STATUS:

6.8 STEUERUNG EINER EPROM-PROGRAMMIEREINRICHTUNG

STATUS:READ

Wie schon in 2.6.3 beschrieben, besteht die Möglichkeit zum

Anschluß einer 2716-EPROM-Programmiereinheit. Mit der Befehlseingabe READ kann der Inhalt eines 2716-EPROMs auf den Bildschirm dargestellt werden. Dabei wird die [CR] Taste für die jeweils folgende Adressen-Darstellung benötigt, während mit einer [ESC] Anweisung der Rücksprung erfolgt. (Ablage der DATEN im PROM-SPEICHER)

STATUS:EPROM

Mit Hilfe dieses Kommandos wird nun ein 2716-EPROM folgendermaßen programmiert:
Die zu programmierenden Daten müssen vom Anwender in den sogenannten PROM-SPEICHER 9000 bis 97FF abelegt werden. Ein systemeigenes Unterprogramm holt sich den Inhalt dieses Speicherbereiches und vollzieht den Programmiervorgang in der Reihenfolge:

STATUS:SET PROM (anschließend: [CR] Taste)

Dies bedeutet den 2716-Baustein in eine vorzusehende Fassung zu stecken und die Programmierspannung von 25V einzuschalten. Nach Beendigung des Programmiervorganges meldet sich das System mit

STATUS:SET PROM,VERIFY

Nun erfolgt eine Prüfung der zuvor einprogrammierten Daten, wobei dem Anwender folgende Mitteilungen übergeben werden:

1. STATUS:SET PROM,VERIFY,NO ERROR

2. STATUS:SET PROM,VERIFY,ERROR!

Die 2. Darstellung bedeutet, daß eine fehlerhafte Speicherzellen-Programmierung erfolgt ist. Ausstieg: [ESC] Taste betätigen.

STATUS:PROMCHECK

Im allgemeinen wird ein EPROM-Baustein vor jedem Programmieren mit Hilfe von UV-Licht in den log."1"-Zustand gebracht. Mit diesem Kommando kann überprüft werden, ob alle 2716-Speicherinhalte den Wert FF besitzen: Ausstieg [ESC] Taste betätigen.

1. STATUS:PROMCHECK,NO ERROR

2. STATUS:PROMCHECK,ERROR!

STATUS:VERIFY

Will man einen 2716-EPROM mit dem CHIP-COMPUTER auf seinen Inhalt hin überprüfen, so erfolgt der VERIFIZIER-Vorgang über den schon erwähnten PROM-SPEICHER 9000H bis 97FFH. Der Anwender hat also vor Kommandoauruf zu sorgen, daß dieser Speicherbereich die zu vergleichenden Daten in der richtigen Weise enthält,siehe auch READ. Auch hier erfolgt eine Mitteilung in der bekannten Weise:

1. STATUS:VERIFY,NO ERROR

2. STATUS:VERIFY,ERROR!

STATUS:LIST

Möchte man einen Ausdruck der 2716-EPROM-Daten erhalten, so geschieht das über die o.a. Befehlsform.
Betrachtet man die einzelnen Befehlsformen genauer, so kann über diese Steuersoftware z.B. auch ein KOPIER-Vorgang vollzogen werden. Dazu steckt man zunächst den EPROM, der die Information enhält, in die Fassung des EPROM-Programmers und überträgt mit dem Kommando: READ seinen Inhalt in den PROM-SPEICHER 9000H bis 97FFH. Anschliessend wird ein gelöschter EPROM eingesteckt und mit Kommando:EPROM kann die Information der Speicheradressen 9000H bis 97FFH in den Baustein übertragen werden.

6.9 ZUSAMMENSTELLUNG SYSTEMEIGENER UNTERPROGRAMME

6.9.1 Unterprogramme zur Bildschirmsteuerung

ADRESSE: ØØCEH Benötigte Register: B,AKKU

Löschen des Bildschirms und CURSOR HOME.

ADRESSE:ØØ92H Benötigte Register: B,AKKU

Ausgabe eines ASCII-Zeichens auf dem Bildschirm, Zeichen muß vor UP-Aufruf im AKKU stehen.

ADRESSE: Ø1Ø6H Benötigte Register: B,AKKU

Unterprogramm für eine CURSOR-Steuerung,dabei können folgende Funktionen gesteuert werden:

1. Ø8H CURSOR links (Zeichen vor UP im AKKU)
2. Ø9H CURSOR rechts
3. ØBH CURSOR nach oben
4. ØAH CURSOR nach unten
5. ØDH RETURN und Löschen bis ZEILEN-Ende
6. 1DH CURSOR setzen (HOME)

ADRESSE: Ø631H Benötigte Register: B,H,L,AKKU

Unterprogramm zur Darstellung einer ASCII-Zeichen-Reihe auf dem Bildschirm.Letztes Zeichen muß C3H sein,wobei der Anfang der Tabelle durch das HL-Registerpaar adressiert wird(ADRESSE-1).

```
BEISPIEL:  LXI H,83FFH        (8400)= 31  ASCII 1
           CALL Ø631H         (8401)= 32  ASCII 2
           HLT                 usw. beliebige ASCII-
                                "             ZEICHEN
                              (844Ø)= C3H
```

6.9.2 Zeitunterprogramme

ADRESSE: 1CBAH Benötigte Register: AKKU

Aufruf eines dezimal wählbaren Zeitunterprogramms mit einem Bereich von Ø bis 9s, durch Belegen des AKKUS im Zehntelsekundenbereich einstellbar.

```
BEISPIEL: MVI A,5ØH;Lege 5s Zeitschleife fest
          CALL 1CBAH;Aufruf UP
          usw.
```

ADRESSE: 1CCAH Benötigte Register: AKKU

Aufruf eines dezimal wählbaren Zeitunterprogramms mit einem Bereich von Ø bis 9Øs, durch Belegen des AKKUS im Sekundenbereich einstellbar.

```
BEISPIEL: MVI A,8ØH;Lege 80s Zeitschleife fest
          CALL 1CCAH;Aufruf UP
          usw.
```

ADRESSE:1CCDH Benötigte Register: AKKU

Aufruf eines HEXADEZ wählbaren Zeitunterprogramms mit einem Bereich von Ø bis 255s, durch Belegen des AKKUS im Sekundenbereich einstellbar.

```
BEISPIEL: MVI A,FEH;Lege 254s Zeitschleife fest
          CALL 1CCDH;Aufruf UP
          usw.
```

6.9.3 Unterprogramme zur Steuerung einer SIEBEN-SEGMENT-ANZEIGE

ADRESSE: 1DØAH Benötigte Register: ALLE

Aufruf eines Unterprogramms, daß es ermöglicht die HEX-ZEICHEN Ø bis F im Anzeigenfeld darzustellen. Dabei wird der Inhalt der Speicherzellen 83F4 bis 84F7 folgendermaßen übertragen:

BEISPIEL: (83F4)=Ø1
(83F5)=23
(83F6)=45
(83F7)=67

Mit dieser Speicherbelegung ergibt sich eine Darstellung auf der Anzeige:

1.SEG 8.SEG
0 1 2 3 4 5 6 7

Die Anzeigenelemente stehen also über die o.a. Speicherzellen "softwaremäßig"in Verbindung. Dabei kann aber noch eine MASKIERUNG über die Speicherzelle 83D9 vorgenommen werden:

(83D9)	BIT 1	steuert	8.	SIEBEN-Segment
	BIT 2	"	7.	"
	BIT 3	"	6.	"
	BIT 4	"	5.	"
	BIT 5	"	4.	"
	BIT 6	"	3.	"
	BIT 7	"	2.	"
	BIT 8	"	1.	"

Bei (83D9)=FF sind alle Anzeigenelemnete angesteuert, während bei (83D9)=ØF nur die Elemente 1. bis 4. aufleuchten.Es muß also vor dem Einstieg in das UP die o.a. Speicherbelegung vorgenommen werden.

```
BEISPIEL:  MVI A,FFH
           STA 83D9H;MASKE für gesamte Anzeige
      OTTO CALL 1DØAH
           JMP OTTO;Bediene laufend die Anzeige
           ORG 83F4H
           DW Ø123H
           DW 4567H
           END
```

ADRESSE: 1D43H Benötigte Register: ALLE

Bei diesem Unterprogramm stehen die Speicherzellen 83F8 bis 83FF mit dem Anzeigenfeld folgendermaßen in Verbindung:

(83F8)-----►1.SIEBEN-SEGMENT
(83F9)-----►2. "
(83FA)-----►3. "
(83FB)-----►4. "
(83FC)-----►5. "
(83FD)-----►6. "
(83FE)-----►7. "
(83FF)-----►8. "

Hierbei lassen sich beliebige Zeichen in der Anzeige realisieren,wobei folgende Zuordnung gilt:

DP g f e d c b a ◄——SEGMENTE
↓ ↓
2^7 2^6 2^5 2^4 2^3 2^2 2^1 $2^Ø$

BEISPIEL:

siehe 2.6.2

6.9.4 Programmierung des USERPORTS-Bausteins 8255A

Bei der Programmierung dieses Bausteins müssen folgende Kanalnummern angegeben werden (siehe auch HARDWARE-Teil).

PORT A	Ø4H
PORT B	Ø5H
PORT C	Ø6H
STEUERWORT	Ø7H

ADRESSE: 3A34H Benötigte Register: AKKU

Unterprogramm zur Umwandlung einer HEX-Zahl in eine ASCII-Zahl. Der umzuwandelnde HEX-Wert muß vor UP-Aufruf in der Speicherzelle BA2AH stehen. Nach Abarbeitung des UP steht die ASCII-Zahl in den Speicherplätzen BA2AH und BA2CH.

BEISPIEL:

```
MVI A,12H;HEX-ZAHL zum Umwandeln im AKKU
STA ØBA2AH
CALL 3A34H;Aufruf des UP
LDA ØBA2AH; Hole ASCII-Zeichen(1)
STA 85ØØH;und speichere ab
LDA ØBA2CH;Hole ASCII-Zeichen(2)
STA 85Ø1H
HLT
END
```

Nach Abarbeitung des Programms haben die Speicherplätze 85ØØ und 85Ø1 folgenden Inhalt:

```
(85ØØ)= 31      (ASCII 1)
(85Ø1)= 32      (ASCII 2)
```

ADRESSE: 3287H Benötigte Register: AKKU

Unterprogramm zur Umwandlung einer ASCII-Zahl in die entsprechende HEX-Zahl. Die umzuwandelnde ASCII-Zahl (2 BYTE) muß vor Aufruf des UP in den Speicherzellen BA2ØH und BA22H stehen. Nach Ablauf des UP steht die HEX-Zahl in BA2ØH.

BEISPIEL:

```
MVI A,31H;(ASCII 1)
STA ØBA2ØH
MVI A,32H;(ASCII 2)
STA ØBA22H
CALL 3287H
LDA ØBA2ØH;Hole HEX-Zahl und lege
STA 85ØØH; auf Adresse 85ØØH.
HLT
END
```

7. EINIGE PROGRAMMIERBEISPIELE

Um den CHIP-COMPUTER zu programmieren, muß nun eine Liste von codierten Befehlen in der richtigen Reihenfolge erstellt werden. Die folgenden, einfachen Beispiele sollen dem Anwender einen kleinen Einblick in das Gebiet der sog. SOFTWARE geben, den PROGRAMMEN, die aus den in richtiger Weise miteinander verbundenen Bauelementen des Systems erst einen funktionierenden Microcomputer machen.
Die Aufgabe des PROGRAMMIERENS besteht nun darin,mit dem gegebenen beschränkten BEFEHLSVORRAT des Microprozessors 8085 eine bestimmte vorgegebene Funktion zu erfüllen. Die zweckmäßige Verwendung der BEFEHLE erfordert Erfahrung und Übung, die am besten mit dem Durcharbeiten von Programm-Beispielen erworben wird.
An dieser Stelle sei auf die weiterführende Literatur, gerade auf dem Gebiet der Programmiertechnnik hingewiesen (siehe auch Literaturverzeichnis).

7.1 AUFGABE

Die Inhalte der Speicheradressen 8ØØØH und 8ØØ1H sollen in die Speicherzellen 8ØØ2H und 8ØØ3H übertragen werden. Dabei sollen aber die Inhalte von 8ØØ2H u. 8ØØ3H gerettet und in die nicht mehr benötigten Speicherinhalte 8ØØØH 8ØØ1H übertragen werden. (Programmtest:STEP-MODE)

7.2 AUFGABE

Die in den Speicherzellen 83ØØH=13 83Ø1H=11 83Ø2H=12 stehenden Werte sollen der Größe entsprechend sortiert werden. (Ergebnis: 83ØØH=13,83Ø1H=12,83Ø2=11). (Programmtest:RUN-MODE)

7.3 AUFGABE

Es sollen die in den Speicheradressen 85ØØH=23 u.85Ø1H=15 abgelegten 8-BIT-Zahlen multipliziert werden, Ergebnis nach 85Ø2H und 85Ø3H. (Programmtest:RUN-MODE)

7.4 AUFGABE

Über den Port A soll ein BINÄR-Zähler auf der LED-Anzeige

dargestellt werden. Ausgehend vom Anfangszustand ØØØØ ØØØØ soll sich der Zähler im Sekunden-Takt bis zu seinem Endzustand 1111 1111 erhöhen.
(Programmtest: STEP-MODE, UP-Zeit dabei nicht verwenden)
(Programmtest: RUN-MODE mit UP-Zeit)

Da sich dieses Manuskript in erster Linie mit der HARDWARE des CHIP-Computers beschäftigen sollte, ist es an dieser Stelle nicht möglich weitere Programmierübungen und Lösungen einzubinden. Hierzu sei noch gesagt, daß durch den praktischen Einsatz dieses 8085-Microcomputersystems bei Aus- u. Fortbildungsmaßnahmen eine große Anzahl interessanter SOFTWARE-Beispiele mit Lösungen entstanden ist, die ev. an einer anderen Stelle veröffentlicht wird.
Ebenso sei darauf hingewiesen, daß sich ein BASIC-Interpreter für den CHIP-Computer in Vorbereitung befindet.

8. LÖSUNGEN

zu 7.1

```
NEW       ORG 8300H
NEW       LHLD 8002H;Hole Inhalt 8002 u.8003
NEW       XCHG;und bringe ins DE
NEW       LHLD 8000H;Hole Inhalt 8000 u.8001
NEW       SHLD 8002H;und speichere in 8002 u.8003
NEW       XCHG
NEW       SHLD 8000H
NEW       ORG 8000H
NEW       DB 10H
NEW       DB 20H
NEW       DB 30H
NEW       DB 40H;Definiere Speicher 8000H bis 8003H
NEW       END
```

zu 7.2

```
NEW       ORG 8500H
NEW ADR1  LXI H,8300H
NEW       MOV B,M
NEW       INX H
NEW       MOV A,M
```

```
NEW      INX H
NEW      MOV C,M
NEW      CMP B; (8301) kleiner (8300)?
NEW      JNZ ADR2
NEW      CMP C; (8301) kleiner (8302)?
NEW      JNZ ADR1
NEW      MOV M,A; (8301) nach 8302
NEW      DCX H
NEW      MOV M,C; (8302) nach 8301
NEW      JMP ADR1
NEW ADR2 DCX H;Inhalt von HL=8301
NEW      MOV M,B; (8300) nach 8301
NEW      DCX H
NEW      MOV M,A; (8301) nach 8300
NEW      JMP ADR1
NEW      ORG 8300H
NEW      DB 13H
NEW      DB 11H
NEW      DB 10H;Definiere Speicher auf Werte
NEW      END
```

zu 7.3

```
NEW      ORG 8200H
NEW      LDA 8500H;Lade Akku mit Multiplikand
NEW      MOV E,A; und bringe ins E-Register
NEW      MVI D,00H;Setze D auf Null
NEW      LDA 8501H;Multiplikator in AKKU
NEW      LXI H,0000H;Hl auf Null
NEW      MVI B,08H;Schleifenzähler
NEW OTTO DAD H;Linksverschieben um eine Ziffer
NEW      RAL;Bits des Multiplikators prüfen
NEW      JNC EMIL; Carry abfragen,Vergleiche mit Hand-Mult!
NEW      DAD D;Multiplikand ins HL hineinaddieren
NEW EMIL DCR B; Alle 8 Bits überprüft?
NEW      JNZ OTTO;Nein dann zurück
NEW      SHLD 8502H; Ja dann speichere Ergebnis ab
NEW      ORG 8500H
NEW      DW 2315H
NEW      END
```

zu 7.4

```
NEW      ORG 8000H
NEW      MVI A,80H;Steuerwort alles Ausgabe
NEW      OUT 07H
NEW      SUB A;Akku auf Null
NEW ADR1 OUT 04H;Inhalt vom Akku auf PORT A
NEW      PUSH PSW;Rette AKKU wegen UP-Zeit
NEW      MVI A,10H;Zeitfaktor
NEW      CALL 1CBAH;UP_Zeit
NEW      POP PSW;AKKU zurück
NEW      INR A;Erhöhe AKKU um 1
NEW      JMP ADR1;und gehe wieder zur Ausgabe PORT.
NEW      END
```

9. BEZUGSQUELLE SYSTEMSOFTWARE u. HARDWARE

Betriebssoftware für CHIP-COMPUTER:
in Form eines Satzes EPROMs + 74S287 programmiert für VIDEO-Interface. PREIS: 180.-(inkl.MwSt.).
Satz Platinen, durchkontaktiert. PREIS: 250.-(inkl.MwSt.).
12K-Byte BASIC-Interpreter , Satz EPROMs. PREIS: 112.-(inkl.MwSt.).
Bausätze u. Zubehör auf Anfrage bei:

Fa. I. Blank
Kantstr. 4
8012 Ottobrunn
Tel. 089/605103

HINWEIS: Die in diesem Manuskript dargestellten Schaltungen des CHIP-Computers, sowie die unter o.a. Adressen zu beziehende SOFTWARE dürfen ausschließlich nur für Amateur- und Lehrzwecke verwendet werden. Eine gewerbliche Nutzung ist nicht erlaubt.

Kurzbeschreibung BASIC-INTERPRETER für den
CHIP-COMPUTER

Der BASIC-INTERPRETER wird mit:
STATUS:WELCOME aufgerufen und belegt den RAM-Speicher von 8000H bis B7FFH.
Ein bereits vorhandenes BASIC-Programm kann mit:
STATUS:WELCOME OLD aufgerufen werden.

Innerhalb des BASIC-INTERPRETERS kann mit folgenden Steuerzeichen von der Tastatur gearbeitet werden:

1. DEL Zeichen löschen
2. ←-- Cursor links und Einfügmodus
3. CONTROL F Zeichen innerhalb einer Zeile löschen
4. --→ Cursor rechts zum Zeilen-Ende
5. ESC Wiederholsteuerung
6. EDIT n Editieren einer Zeilennummer,dann ESC
7. CONTROL S Anzeige stoppen,dann beliebige Taste für Weiterlauf
8. CONTROL B Ausstieg aus BASIC,man kann mit WELCOME OLD zurück
9. ESC nach RUN hält den BASIC-Interpreter an.

Der Interpreter hat Gleitpunktarithmetik und Stringbehandlung, er arbeitet mit sechs Stellen im Bereich von -10^{38} bis $+10^{38}$. Die maximale Zeichenanzahl einer Zeile (64) beschränkt Stringvariable. Funktionen können definiert werden,Felder werden durch die Anzahl der Speicherplätze begrenzt.
Programme werden mit Zeilennummern eingegeben, die als Sprungmarken verwendet werden.Befehle die sofort ausgeführt werden sollen, werden ohne Zeilennummern angegeben:

```
10 PRINT"CHIP-COMPUTER"      PRINT"CHIP-COMPUTER"
20 END
RUN
```

Zeilennummern können nicht doppelt vorkommen.Wird eine Zeilennummer zum zweitenmal verwendet,so wird der alte Inhalt überschrieben.
Befehle innerhalb einer Zeile können mit ":" getrennt werden.

Befehlsübersicht für den BASIC-INTERPRETER:

1. LIST	Ausgeben eines BASIC-Programms auf Bildschirm,Zeilennummer kann als Anfangswert angegeben werden.
2. NEW	Lösche Programm und den Arbeitsbereich.
3. RUN	Starte ein geladenes Programm
4. CONT	Ist eine Programm auf ein STOP gelaufen, so kann mit dem nächsten Befehl weitergearbeitet werden.
5. CLEAR	Felder und Variable werden gelöscht.
6. DATA	Datenangaben,die im System gespeichert werden und später mit READ gelesen werden.
7. READ	Bezieht sich auf DATA.
8. REM	Kommentar-Zeile
9. STOP	Der Interpreter hält an u. kann mit CONT weiterlaufen.
10 RETURN	Rücksprung von einem Unterprogramm nach GOSUB.
11.GOSUB	Sprunganweisung zu einem Unterprogramm mit Zeilennummer.
12.GOTO	Unbedingter Sprung
13.IF	Bedingte Anweisung
14.FOR	Programmieren einer Schleife.
15.NEXT	"
16.STEP	Schrittweite innerhalb einer Schleife.
17.ON	Bedingte Sprunganweisung.
18.LET	Zuweisung
19.INPUT	Daten, die von der Tastatur eingelesen werden sollen
20.POKE	Daten in den Speicher einschreiben
21.PEEK	Inhalt einer Speicheradresse
22.THEN	Bezieht sich auf IF.
23.TO	Laufbereich für Schleifenvariable
24.DEF	Definition einer Funktion
25.DIM	Felder werden definiert.
26.PRINT	Ausgabe von Variablen oder Strings.(";" kein Abst."," mit)
27.RESTORE	Zeiger wird auf DATA zurückgesetzt.
28.TAB	Zeiger für Ausgabe wird gesetzt.
29.SPC	Leerzeichen einfügen.
30.POS	liefert Ausgabeposition(Spalte in einer Zeile)
31.FN	Aufruf einer Funktion (vorher durch DEF festgelegt)
32.FRE	liefert den noch verfügbaren Speicherplatz(PRINT FRE(0))
33.INP	Daten von einem 8255 PORT lesen (A=INP(04)

34.math.Funktionen:

* Multiplikation
/ Division
↑ Potenzieren
+ Addition
- Subtraktion
< Kleiner
> Größer
= Gleich

EXP Exponentialfunktion
INT Ganzzahlige Darstellung
SGN Signum
RND Zufallsgenerator
ABS Absolutbetrag

COS Kosinus
TAN Tangens
ATN Arcus-Tangens
SQR Quadrat-Wurzel
SIN Sinus
LOG Logarithmus n

NOT Log.Negation
OR Log. ODER
AND Log. UND

35. Stringfunktionen:

ASC ASCII nach DEZIMAL
CHR$ DEZIMAL nach ASCII
LEN Anzahl String-Zeichen
VAL String in Wert String
LEFT$ Linker Teil eines Strings
RIGHT$ Rechter Teil eines Strings
MID$ Mittelteil eines Strings
STR$ Dezimal in String

Basic-Programme abspeichern/laden über: SAVE"R/LOAD"R

SAVE"R bedeutet ein Basic-Programm auf Kassette abspeichern.

LOAD"R bedeutet ein Basic-Programm von der Kassette einlesen.

In beiden Fällen kann mit der ESC-Taste der Aus-bzw. Einlesevorgang unterbrochen werden.

36. Aufruf eines ASSEMBLER-Programms über: USR-Funktion

Es können innerhalb eines BASIC-Programms Unterprogramme über "USR" (USER SUBROUTINE) aufgerufen werden, die im Maschinencode 8085 vorliegen. Dabei sollte aber folgendes beachtet werden:

1. STATUS:EDIT NEW löscht ein BASIC-Programm!

2. STATUS:EDIT OLD löscht ein BASIC-Programm nicht!

Da der BASIC-Interpreter den RAM-Bereich von:

8ØØØH bis B7FFH belegt

sollten die Unterprogramme von Adresse:

6ØØØH im Maschinencode

abgelegt sein.

Beispiel:

```
10 PRINT " TEST MASCHINENCODE"
20 B=24576
30 A=USR(B)
40 PRINT " ERFOLGREICH"
50 END
```

BASIC

```
      ORG 6000H
OTTO  IN 09H
      RLC;WARTE AUF TASTE
      JNC OTTO
      RET
      END
```

ASSEMBLER

37. Steuerung der Ausgabeform Bildschirm/Drucker über: ASG1,ASG2

Beispiel: ASG1:LIST — Zeichen werden nur dem Drucker übergeben.

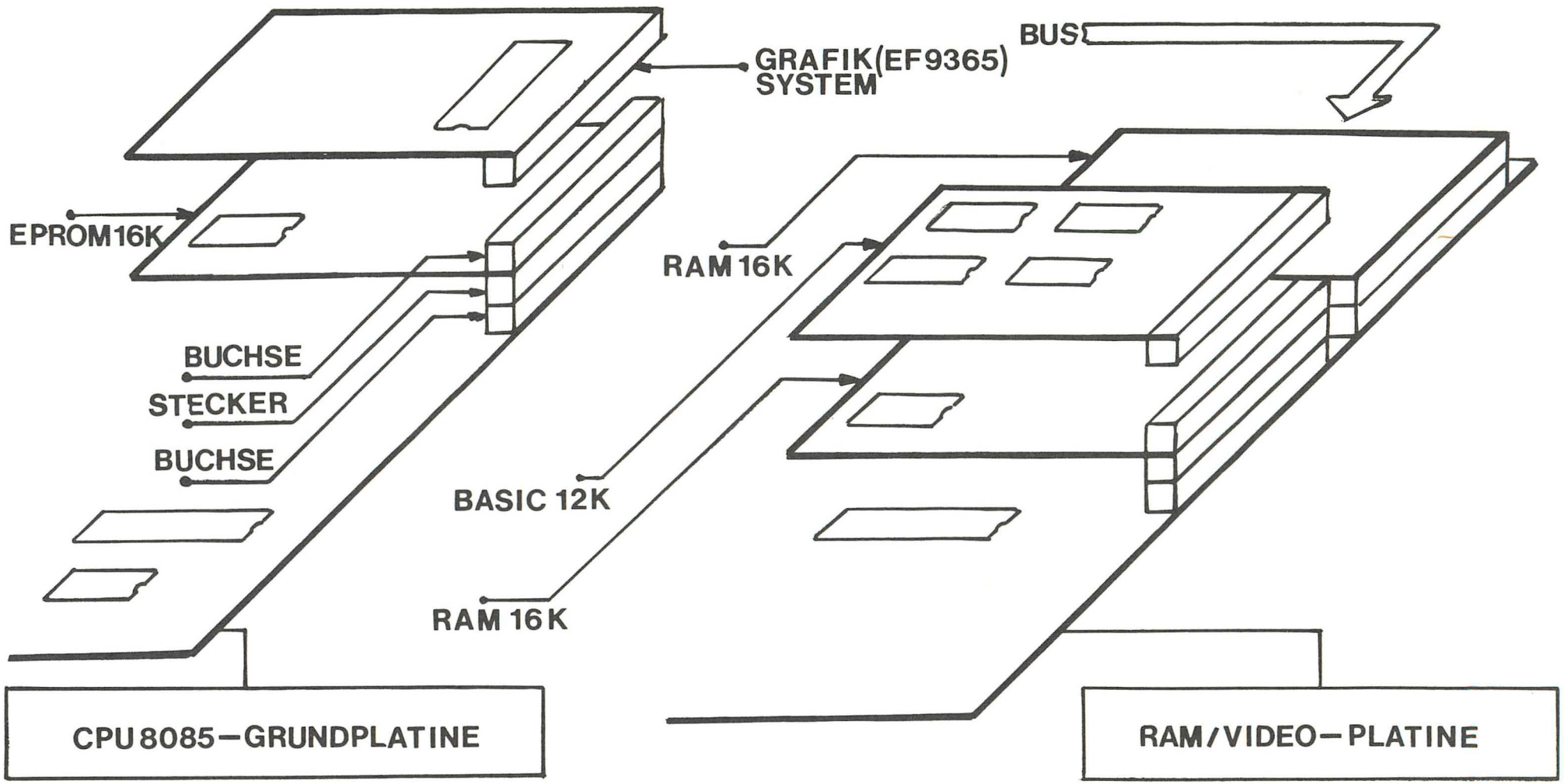
ASG2:LIST — Zeichen werden dem Drucker und dem Bildschirm übergeben.

Beispiel:

```
10 ASG1
20 PRINT"TEST"
30 ASG2
40 PRINT"BASIC CHIP_COMPUTER"
```

Fehlermeldungen:

SN	SYNTAX ERROR--------	Das Befehlsformat ist nicht beachtet worden.
FC	FUNCTION----------	Innerhalb der Funktion liegt ein Parameter außerhalb des zulässigen Bereiches.
OD	OUT OF DATA---------	Mehr READ Anweisungen,als mit DATA bereitgestellt.
BS	BAD SUBSCRIPT-------	Ansprache eines Matrix-Elementes,das außerhalb der Matrix-Dimensionierung liegt.
TM	TYP MISMATCH--------	Behandlung von Variablen als String oder umgekehrt.
OS	OUT OF STRING-SPACE-	Der zur Verfügung stehende String-Speicher ist durch zuviele Strings belegt.
LS	LONG STRING---------	Zeichenkette länger als 255 Zeichen.
ST	STRING TOO COMPLEX--	Aufteilen in mehrere Strings,der vorhandene String ist zu komplex.
UF	UNDEFINED FUNCTION--	Eine vorher nicht definierte Funktion aufgerufen.
CN	CAN`T CONTINUE------	Mit CONT ist eine Programmfortsetzung nicht möglich,da ein Fehler vorliegt.
ID	ILLEGAL DIRECT------	Anweisung ist als Direkt-Kommando nicht möglich.
US	UNDEFINED STATEMENT-	Sprungziel existiert nicht.
DD	DOUBLE DIMENSION----	Mehrfach-Definition eines Feldes.
RG	RETURN WITHOUT GOSUB------	Unterprogramm ohne GOSUB.
/O	DIVISION BY ZERO----	Division durch Null verboten.
OV	OVERFLOW----------	Der zulässige Zahlenbereich ist überschritten worden.
OM	OUT OF MEMORY-------	Der vorhandene Speicherplatz reicht nicht mehr aus.
NF	NEXT WITHOUT FOR----	Bei einer Programmschleife fehlt die FOR-Anweisung.



CHIP—COMPUTER

LITERATUR-VERZEICHNIS

1. Datenbuch 8085 SIEMENS AG München
2. Datenbuch Speicher SIEMENS AG München
3. Applikations-Schrift THOMSON EF 9364

4. uP-Hobby Band 1:Mikroprozessor 8085
   für den Anwender,Herbert Bernstein

5. 8080/8085 Programmieren in ASSEMBLER
   L.A. Leventhal

6. uP-Hobby Band 2:Vom Chip zum System
   Herbert Bernstein